



HAL
open science

Human motion transfer on humanoid robot.

Francisco Javier Montecillo Puente

► **To cite this version:**

Francisco Javier Montecillo Puente. Human motion transfer on humanoid robot.. Automatic. Institut National Polytechnique de Toulouse - INPT, 2010. English. NNT: . tel-04278718v1

HAL Id: tel-04278718

<https://theses.hal.science/tel-04278718v1>

Submitted on 23 Nov 2010 (v1), last revised 10 Nov 2023 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ÉCOLE DOCTORALE SYSTÈMES

THÈSE

pour obtenir le grade de

Docteur de l'Université de Toulouse
délivré par Institut National Polytechnique de Toulouse

Spécialité: Systèmes Automatiques
présentée et soutenue publiquement le 26 août 2010

Transfert de Mouvement Humain vers Robot Humanoïde Human Motion Transfer on Humanoid Robot

Francisco Javier Montecillo Puente

Préparée au Laboratoire d'Analyse et d'Architecture des Systèmes
sous la direction de M. Jean Paul Laumond

Jury

Mme. Aude BILLARD	Rapporteur
M. Armel CRETUAL	Rapporteur
Mme. Katja MOMBAUR	Examineur
M. Frédéric LERASLE	Examineur
M. Jean Paul LAUMOND	Directeur de Thèse

Abstract

The aim of this thesis is to transfer human motion to a humanoid robot online. In the first part of this work, the human motion recorded by a motion capture system is analyzed to extract salient features that are to be transferred on the humanoid robot. We introduce the humanoid normalized model as the set of motion properties. In the second part of this work, the robot motion that includes the human motion features is computed using the inverse kinematics with priority. In order to transfer the motion properties a stack of tasks is predefined. Each motion property in the humanoid normalized model corresponds to one target in the stack of tasks. We propose a framework to transfer human motion online as close as possible to a human motion performance for the upper body. Finally, we study the problem of transferring feet motion. In this study, the motion of feet is analyzed to extract the Euclidean trajectories adapted to the robot. Moreover, the trajectory of the center of mass which ensures that the robot does not fall is calculated from the feet positions and the inverse pendulum model of the robot. Using this result, it is possible to achieve complete imitation of upper body movements and including feet motion.

Résumé

Le but de cette thèse est le transfert du mouvement humain vers un robot humanoïde en ligne. Dans une première partie, le mouvement humain, enregistré par un système de capture de mouvement, est analysé pour extraire des caractéristiques qui doivent être transférées vers le robot humanoïde. Dans un deuxième temps, le mouvement du robot qui comprend ces caractéristiques est calculé en utilisant la cinématique inverse avec priorité. L'ensemble des tâches avec leurs priorités est ainsi transféré. La méthode permet une reproduction du mouvement la plus fidèle possible, en ligne et pour le haut du corps. Finalement, nous étudions le problème du transfert mouvement des pieds. Pour cette étude, le mouvement des pieds est analysé pour extraire les trajectoires euclidiennes qui sont adaptées au robot. Les trajectoires du centre de masse qui garantissent que le robot ne tombe pas sont calculées à partir de la position des pieds et du modèle du pendule inverse. Il est ainsi possible de réaliser une imitation complète incluant les mouvements du haut du corps ainsi que les mouvements des pieds.

Acknowledgments

It is a pleasure to thank to many people who made this thesis possible.

First, I would like to thank Jean-Paul Laumond. It is difficult to state my gratitude to him because of his patience and advice during these long three years.

I wish to express my sincere gratitude to Aude Billard, Armel Cretual, Katja Mombaur and Frédéric Lerasle, for agreeing to be part of my thesis committee and for their valuable suggestions for the improvement of this work.

I gratefully acknowledge the Mexican National Science and Technology Counsel (CONACyT) for its financial support during my stay in France. My work has been partially supported by the French project ANR-PSIROB LOCANTHROPE.

Special thanks to all the members of Gepetto team because of their support and friendship, Manish, Anh, Sebastien, David, Ali, Minh, Mathieu, Oussama, Thomas, Duong, Layale, Sovan.

I would like to thank to the people who have provided their assistance at LAAS, Anthony Mallet and Marc Vaisset.

Thanks to the colleagues and friends at LAAS, specially, Luis, Juan Pablo, Mario-Dora, Ixbalam, Jorch, David, Ali (Pakistan), Hung, Thanh, Diego, Gilberto, Xavier.

I also thank to Carlos, Vero, Marlen, Edith, Jose Luis, Manuel, Memo, Sebastien H. and his gang, because all these days of fun and friendship.

I will never forget the moments enjoyed with my roommates Carlos, Diego and Anh.

I would like to express special thanks to Doc. Víctor Ayala and Raúl Sánchez whom always encourage me to reach my goals.

I wish to thank to my entire extended family for providing a loving environment for me. My brothers and sisters (Caro, David, Noe, Viki) were particularly supportive.

Finally, I want to thank to two very special persons Laura and Evelyn.

Contents

1	Introduction	7
1.1	Measuring Human Movement	7
1.2	Humanoid Robots	10
1.3	State of the Art	12
1.3.1	Context on Humanoid Motion	14
1.3.2	Computer Animation	15
1.3.3	Human Motion Imitation by Experimental Platforms	16
1.4	Problem Statement	21
1.5	Contribution	22
1.6	Thesis Outline	23
1.7	Publication List	24
2	Inverse Kinematics and Tasks	25
2.1	Inverse Kinematics	27
2.2	Inverse Kinematics: Solving Two Tasks	31
2.3	Inverse Kinematics: Solving Multiple Tasks with Priority	32
2.4	Damped Inverse Kinematics	33
2.5	Nonlinear Programming: Inverse Kinematics, Tasks and Constraints	34
2.6	Framework for Control of Redundant Manipulators	35
2.7	Conclusions	36
3	Human Motion Representation	37
3.1	Humanoid Normalized Model	40
3.2	Humanoid Normalized Model Representation	46
3.3	Model Validation	47
3.3.1	Human Motion Data: Skeleton and Markers	47
3.3.2	Skeleton Animated by Humanoid Normalized Model	49
3.4	Conclusion	51

4	Online Human-Humanoid Imitation	57
4.1	Upper Body Imitation by HRP2-14	57
4.2	Seamless Human Motion Capture	59
4.3	Whole Body Motion Generation	61
4.4	Center of Mass Anticipation Model	62
4.5	Tests and Evaluation	63
4.5.1	Practical Implementation	63
4.5.2	Dancing	64
4.5.3	Foot Lift	65
4.5.4	Quality of motion imitation	68
4.5.5	Transfer Motion Limitations	68
4.6	Conclusion	71
5	Imitating Human Motion: Including Stepping	76
5.1	Offline Human Feet Motion Imitation	78
5.1.1	Feet Motion Segmentation	78
5.1.2	Planning Feet Motion	81
5.1.3	Planning Zero Moment Point and Center of Mass	82
5.1.4	Transfer Feet Motion to Humanoid Robot	85
5.2	Online Human Feet Motion Imitation	85
5.2.1	Human Motion Segments	87
5.2.2	Planning Feet Motion	88
5.2.3	Planning Center of Mass Motion	91
5.2.4	Transfer Feet Motion to Humanoid	91
5.3	Imitation Including Stepping	91
5.3.1	A Complete Humanoid Normalized Model	93
5.3.2	Human Motion Transfer Including Stepping	93
5.4	Conclusion	93
6	Conclusion and Perspectives	95
6.1	Conclusion	95
6.2	Perspectives	97
A	Motion Capture and Marker Set for our Experiments	99

1

Introduction

The study of human movement concerns itself with understanding how and why people move and the factors that limit or enhance our capability to move. This includes fundamental skills used on a daily basis (such as walking, reaching and grasping), advanced skills (like sport and dance), exercising for health or rehabilitation of an injured limb. In parallel to the study of human movements is the field of humanoid robotics where a lot of research goes into improving robot autonomy. Motion generation to achieve human-like, reliable and safe motion for humanoid robots is among the current challenging problems in robotics. In this context, the work in this thesis is formulated around the key-points of human motion and online motion generation for humanoid robots.

1.1 Measuring Human Movement

Kinesiology is defined as the scientific study of human movement. To assess human motion, Kinesiology involves principles and methods from biomechanics, anatomy, physiology and motor learning. Its range of application includes health promotion, rehabilitation, ergonomics, health and safety in industry, disability management, among others. The measurement of human movement is one of the tools that is central in this research field. For this reason we present a short review on the development of human motion measurement and analysis in the following paragraphs. This summary is based mainly on the studies of [Rosenhahn et al. 2008; Medved 2001; Baker 2007].

Humans have been expressing the sense of motion in written/pictographic forms since a very long time. One of the earliest examples of this can be seen in the cave painting of a running bison (Fig. 1.1). The sense of motion in the bison was expressed by the multiplicity of its legs. This particular painting was discovered in Chauvet-Pont-d'Arc Cave in Ardèche France which dates back to 32000 years ago. Some theories state that the utility of such paintings were to communicate between prehistoric people, while others argue that they were used for religious and ceremonial purposes. These kinds of pictures representing motion reveal that since prehistoric days motion has been a subject of interest.



Figure 1.1: Since prehistoric days motion was a subject of interest. In this picture the sense of dynamics of a running bison is depicted by the multiplicity of legs, Jean-Marie Chauvet ©DRAC.

In Ancient Greece, studies on motion were also conducted. Particularly, Aristotle (383 B.C. to 321 B.C.) published texts about gait in animals which included some observations about motion patterns involved in humans. In the Renaissance period, Leonardo da Vinci (1452-1519) stated that it was indispensable for a painter to become familiar with anatomy to understand which muscles caused particular motions of the human body parts. Furthermore, a detailed description about how humans climb stairs was also given (see Figure 1.2). In those days, art was a discipline which devoted a lot of effort to studying human motion. Alfonso Borelli (1608-1679) was one of the pioneers on the measurement and analysis of human locomotion from a quantitative point of view. The foundation of modern dynamics was laid down by Isaac Newton during the “Enlightenment period”. The three laws of motion, were a very crucial contribution to understanding human motion. From an analytical point of view they also achieved more accurate results than any previous methods.

In the 19th century, a variety of devices were built to produce moving pictures. Particularly, chronophotography was the most advanced technique for the study of movement. Muybridge

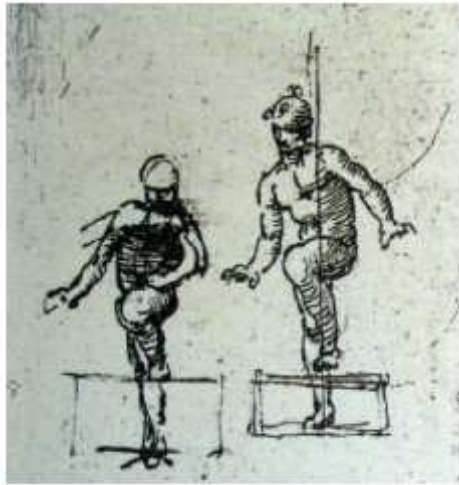


Figure 1.2: Painting of a human climbing stairs. The cross mark at the right foot indicates the contact point used to reach the current posture, the vertical line expresses the center of mass ground projection. Da Vinci was interested in studying the magnitude of the force at contact points.

was one of the founders in recording fast motions, notably the recording of a galloping horse in 1878. Motion studies of Muybridge based on multiple images are often cited in the context of the beginnings of biomechanics. A popular reference picture is the walking sequence of a man (Figure 1.3). The stance phases during a cycle of walking and the leg motion are the main interests of this kind of a sequence. The white marks in the background of the picture help to measure the stride of the step.

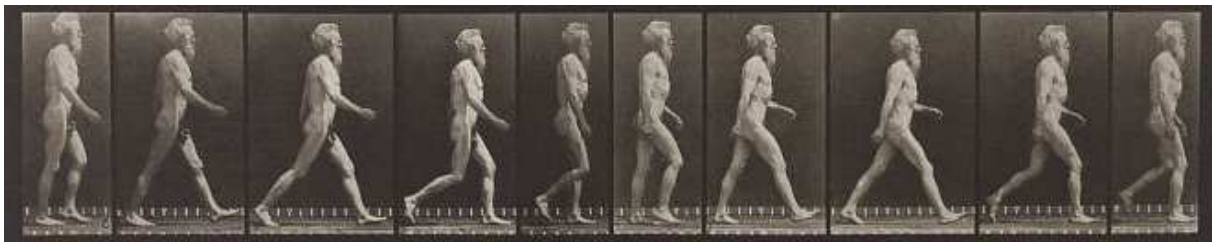


Figure 1.3: The cycle of locomotion is clearly shown in this sequence of a man walking: the single foot support stance, the double support stance and the transitions between. The coordination between arms and legs is also clearly observed. Picture from [Muybridge 1979].

Among other major contributions in motion study in this century was the work conducted by the Weber brothers [Weber and Weber 1991]. Mainly, they studied the path of the center of mass during human displacement and focussed on the computation of human walking using differential equations. The last study is the precursor to today's computer graphics animation. In Figure 1.4, we show an example picture of computed human motion. Etienne-Jules Marey, a

French physician and physiologist, also contributed to the development of recording devices, in particular the odograph.



Figure 1.4: The Weber Brothers computed the evolution of the human locomotion by solving differential equations [Weber and Weber 1991]. In this computed picture we can observe the movement of the leg for one step.

The first experimental studies of human gait, i.e. determining physical quantities like inertial properties, were conducted by Christian W. Braune (1831-1892) and Otto Fischer (1861-1917). They considered the human body as rigid bodies in form of dynamic links in series. The work of Nicholas Bernstein (1896-1966) in Moscow introduced the 3D analysis based on cameras. The methods for measuring human movement continued improving with the advent of new technologies like electronics and magnetic devices, up until today's motion capture systems based on reflective markers, magnetic or inertial devices.

1.2 Humanoid Robots

The idea of building machines which look and move like humans has been explored by philosophers and mathematicians since antiquity. Nowadays, the concepts of such machines are a part of research in robotics. Humanoid robots can be thought of as mechanical, actuated devices that can perform human-like manipulation, including locomotion as their main skill for displacement. Well before the first modern humanoid robot, one of the biggest steps towards this objective was achieved in 1956 with the first commercial robot manipulator, Unimate, from Unimation. The automotive industry was the first to benefit from these kinds of manipulator robots. They were used to achieve systematic tasks like welding, spraying paint, cutting, picking and placing objects etc. In parallel with a large amount of research interest, the first humanoid robot

was revealed by the Japanese company Honda. The unveiling of the humanoid robot platform P2 in 1996 (Figure 1.5) surprised the robotics community, and P2 was henceforth the reference picture of a humanoid robot.

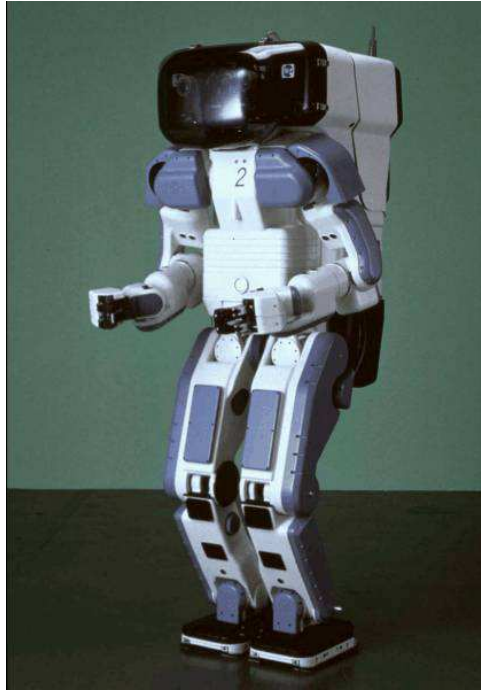


Figure 1.5: P2: the Honda humanoid presented in 1996 that was capable of walking and climbing/descending of stairs. It is 210 kg weight and 182 cm height. Picture source <http://asimo.honda.com>

Current goals of research in humanoid robotics include industrial applications of humanoid robots and their inclusion into daily human lives. Tanie, [Tanie 2003], conducted a study to determine the range of applications of humanoid robots, here we briefly recall the main results,

- maintenance tasks of industrial plants,
- security services for home and office,
- human care, teleoperation of construction machines,
- cooperative work.

Additionally, a survey was carried to inquiry how the human body appearance can be utilized in practice. The main results were:

- the behavior of a human-like robot produces emotional feelings useful for friendly communication between robot and human,
- human shape is one of the best shapes for a remotely controlled robot,
- there are many cases where human-shaped devices can replace the human worker.

Since then many studies have explored this aspect of robot motion and how to make robots 'friendly' and 'human-aware'. Human-robot interaction is now a challenging research field and further user studies on the efficacy of humanoid robots in human environments can be found in [Rich et al. 2010] [Saint-Aime et al. 2009] [Marin-Urias et al. 2009].

Some examples of applications developed on the humanoid robot HRP-2 includes the displacement of large objects by using a pivoting technique (Fig. 1.6). The humanoid plans its motion using the random probabilistic methods to explore all the possible paths to attain the objective location [Yoshida et al. 2007]. In Figure 1.7 the robot carries a large and cumbersome barbell in its hands. The goal of the robot is to carry the object from one location to another while avoiding collision with the obstacles presented in the working place [Esteves Jaramillo 2007]. An application of interaction between humans and HRP-2 via speech is shown in Figure 1.8. The user gives instructions that the robot can interpret. These instructions help guide the robot to locate objects, walk to a table, to pick and to release objects. In the Figure 1.9 the humanoid being teleoperated is shown [Peer et al. 2008]. This demonstration was taking place by teleoperating from two distant laboratories, one located in Germany and other in Japan.



Figure 1.6: From [Yoshida et al. 2007]: HRP-2 is displacing a big wooden box from one location to another by pivoting the object. In this example the motion is precomputed in advance, the locomotion being decoupled from the manipulation.

1.3 State of the Art

Our work deals with human-humanoid motion imitation, this topic is part of a more general one that deals with humanoid robotics.



Figure 1.7: From [Esteves Jaramillo 2007]: the robot HRP-2 is carrying an object from one place to another while avoiding collisions between its own body, the manipulated object and the other objects in the environment.

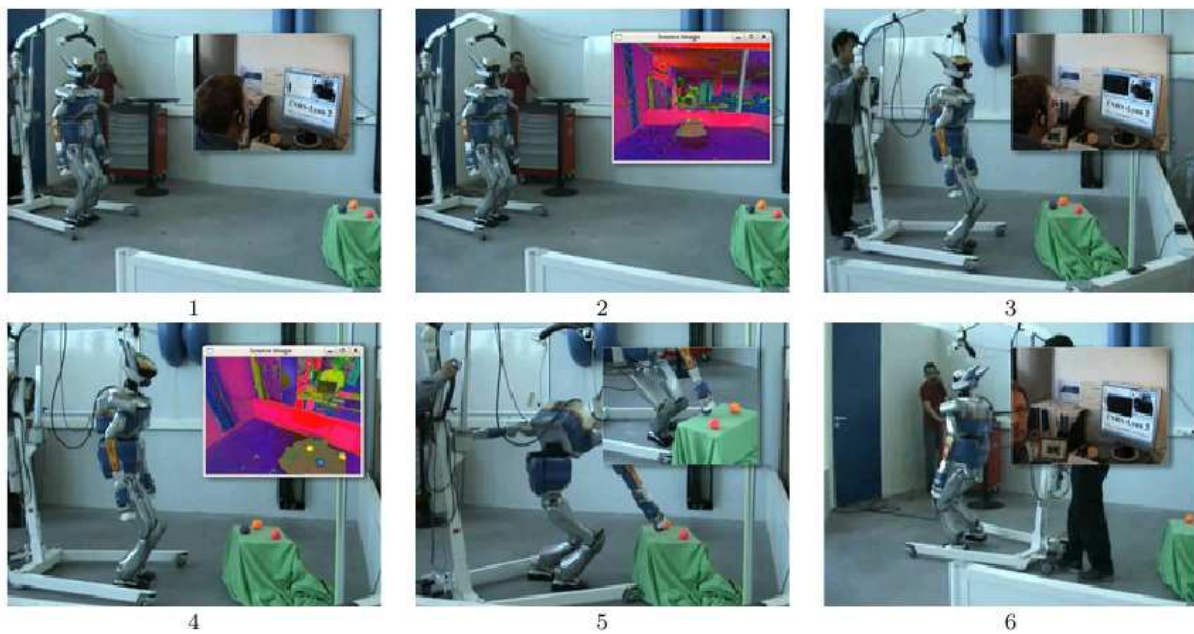


Figure 1.8: From [Yoshida et al. 2007]: the robot is being commanded by speech to locate, grasp and release a colored ball, or even to walk to some visually identified locations.

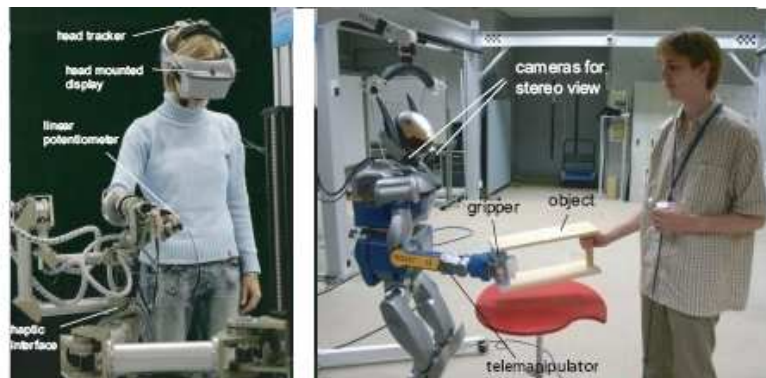


Figure 1.9: From [Peer et al. 2008]: HRP-2 is being teleoperated from Germany to Japan. The objective is to assist a person to grasp and release an object. The left image shows the human operator and the human-system-interface, while the right image shows HRP-2 collaborating with a human at the remote site.

1.3.1 Context on Humanoid Motion

The first studies on humanoid robots aimed at building robots capable of walking, however the main difficulty to overcome was dynamic balance. The first viable formulation that gave us insights into balance was proposed by Vukobratovic [Vukobratovic and Stepanenko 1972]. One of the major contributions was the concept of the Zero Moment Point (ZMP) as criteria to determine dynamic balance for legged robots was introduced. It specifies the point with respect to which dynamic reaction force at the contact of the foot with the ground does not produce any moment, i.e. the point where total inertia force equals zero. The concept assumes the contact area is planar and has sufficiently high friction to keep the feet from sliding. The formulation and calculation of the ZMP is further detailed in Chapter 4. Advanced robot platforms like the H6 [Kagami et al. 2001], ASIMO [Sakagami et al. 2002], QRIO [Ishida et al. 2003], HRP-2 [Kaneko et al. 2004], HUBO [Ill-Woo et al. 2005], HOAP3 [Zaier and Kanda 2007] WABIAN-[Ogura et al. 2006] and Mahru III [Woong et al. 2007] maintain balance through manipulating their ZMP via the Center of Mass (CoM) location or by directly modifying the COM position.

In parallel to humanoid robots is the field of animating virtual anthropomorphic figures or avatars. As the computation power of computers increased, animation companies harnessed this towards realistic and complex character behaviors. However, this is not always a straightforward affair. Animation using computers became part of the film industry with the release of the Japanese film *Final Fantasy* in 2001. This film was the first one to animate characters using the novel computer animation techniques based on motion capture technology. Since then computer animation has come a long way with modern cutting edge technologies showcased in films like *Avatar* (2009).

Both humanoid robot and computer animation research groups are constantly developing methods for generating motions for their robots and characters, respectively. These methods

vary from synthesized motions to motion learning. In the rest of this section, we review some of these.

1.3.2 Computer Animation

Motion capture technology is the most reasonable choice when high quality motions for computer animation purposes has to be produced in a short period of time. The captured data is specific to the character of the person performing the actions. If the data is to be reused to animate another character, it has to be “retargeted” to account for the differences in the structure.

The space-time constraint method was proposed by Gleicher to deal with the retargeting problem using captured data, Figure 1.10 [Gleicher 1998]. This method minimizes an objective function subject to equality constraints. The objective function is defined as the time integral of the difference between the original and target motions, while the constraints represent spatio-temporal relations between the body segments and the environment. Also, these constraints are used to codify locations at specific time instants of predefined tags attached to the character. A scaling phase is used when the height difference between characters is considerable. The matching measure between characters was encoded in the objective function. The main drawback of this method is its inherently offline nature because the whole motion sequence must be known in advance.

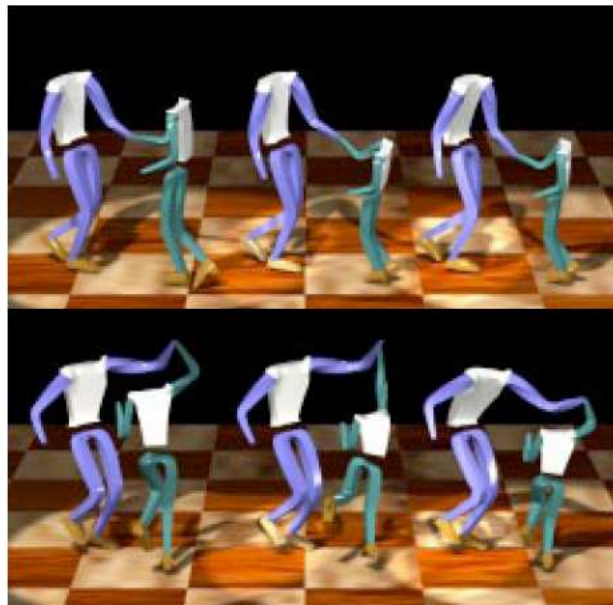


Figure 1.10: Adapted from [Gleicher 1998]. The left image shows the original dance motion. In the center image only motion of the female character, whose size has been modified, is adapted (at the right of the center image). In the right image both character sizes have been modified, hence the motion is adapted to both dancers.

The authors in [Choi and Ko 2000] used motion capture data for online motion retargeting to a new character. The method proposed to solve the retargeting problem is based on a prioritized inverse kinematics solver which allows to solve several goal with predefined priority. The main goal of this method is to track specified end-effector trajectories, i.e. position and orientation of rigid bodies representing head, hands and feet. The secondary goal was to track joint angles of the original character. The anthropometric differences between character is managed by the end-effector trajectories, while the motion properties were encoded in the secondary goal.

A flexible framework for animating virtual characters is the application MKM, [Multon et al. 2008]. In this work the authors proposed a motion representation which is independent from the character's morphology. It is based on both Cartesian and angular data, Figure 1.11. The key contribution of this work is a representation which decomposes the character's skeleton in a way that allows the use of an analytical solution of the inverse kinematics problem. The representation is a normalized skeleton that allows retargeting between different characters.

The normalized skeleton is divided into three parts: normalized segments, limbs of varying length, and the spine. The normalized segments are defined by one body segment including hands, feet and hips. These codify their Cartesian positions in the reference frame. The limbs with variable length include the upper and lower limbs. Here, the hypothesis is that limb articulations are contained within planes. The representation of the spine is expressed by a spline function. The main advantages of this representations are: fast computation of the retargeting, it is reversible, only some trajectories are recorded. Some of the disadvantages of this representation are: the skeleton hierarchy is predefined, it is required specialized inverse kinematic solver.

1.3.3 Human Motion Imitation by Experimental Platforms

Direct motion imitation

Generating motion for real humanoids from captured data is more challenging, because physical limits and balance must be consciously taken into account. The good reference in this regard is the work conducted by Nakaoka, [Nakaoka et al. 2003]. Here, the humanoid robot HRP-2 was able to execute the famous and visually striking, traditional Aizu-Bandaisan Japanese dance, Figure 1.12. The authors in this study generated upper body motion for the HRP-2 robot using motion capture markers on human dancers and inverse kinematics. The leg motion of the dancers was analyzed and extracted as motion primitives. The upper and lower body motions were then coupled and modified such that the computed robot motion satisfied dynamic stability criteria, the Zero Moment Point (ZMP). This method was implemented offline and required several runs of the process to reach a viable solution.

Imitation using optimization

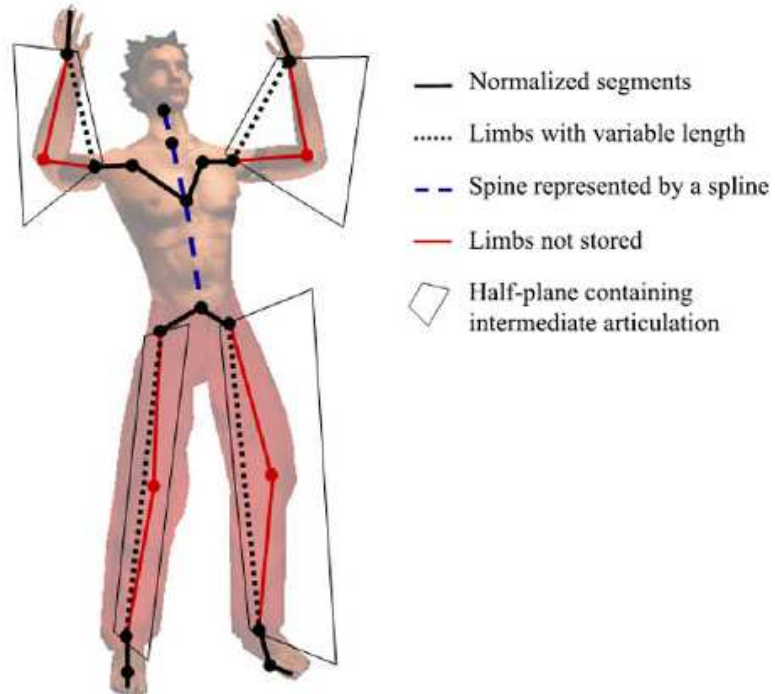


Figure 1.11: MKM: a skeleton representation independent of the character body, adapted from [Multon et al. 2008].



Figure 1.12: Aizu-Bandaisan dance performed by HRP-2, adapted from [Nakaoka et al. 2003].

Optimization techniques has been used to solve the problem of generating motion taking into account for the physical limits of the robot by defining appropriate constraints. Ude et. al proposed to solve a large scale optimization problem to generate joint trajectories for a DB robot [Ude et al. 2004]. Joint angle trajectories of human motion were computed by embedding a scalable kinematic structure to the captured human motion. Robot balance was not taken into consideration. In the study by Ruchanurucks et al. [Ruchanurucks et al. 2006], a non-linear optimization problem was also solved subject to joint limits, autocollision, velocity limits and force limits constraints. To increase the convergence speed the motion was parametrized by B-splines. In Suleiman et. al [Suleiman et al. 2008], first the joint motion data was scaled onto the humanoid robot's joints. Then, an optimization problem was solved to fit this motion to the robot structure and its physical limits, keeping in mind the analytical gradient of the dynamic model. In these studies human motion was represented by joint trajectories and some specific Cartesian positions of points of the human body.

Machine Learning Approaches

Other approaches of motion generation are based on machine learning algorithms. Mainly, a great deal of attention has been focused on the behavior acquisition methods using imitation of human behavior. Schaal [Schaal 1999] discussed the question of whether imitation learning was a way for learning. In the same way, imitation learning in [Guenter et al. 2007] is used to approach the problem of imitating constrained reaching movements. The problems to solve in this method are “what to imitate” and “how to imitate”. The former consists of extracting the principal constraints that appears in demonstrated tasks, and the later was related to finding solutions that reproduces the demonstrated task in different situations. This method used reinforcement learning (RL) to learn its own ways of acting, aiming to improve its abilities. A reward function was used in the learning phase, and represented the similarity between the current trajectory and the one modulated by the learning system. If in the current trial the system fails to reach the target, the modulation was changed by reinforcement learning. This method used joint trajectories, so even if the end-effector task had been reached the target, the joint trajectories had important differences. It is observed that the trajectories in the joint space varies significantly, while the hand trajectory remains quite constant [Billard et al. 2006].

Inamura et al. developed a system that is capable of abstracting others' behaviors, recognize behaviors and generate motion patterns through the mimesis framework [Inamura et al. 2001]. The mimesis loop consists of first segmenting continuous human motion patterns into 'self-motions'. These self-motions are then transformed into primitive symbols by an abstraction process. Each such symbol can be considered as a behavior. The information encoded into self motions are joint trajectories segments. Hidden Markov Models (HMMs) were used to represent relations between motion patterns and primitive symbols. To synthesise new motion a sequence of self motion elements was first built from primitive symbols. Finally, these

were transformed into joint angles trajectories while considering dynamic conditions [Yamane and Nakamura 2003]. This approach has been used to communicate between a human and a humanoid robot [Takano et al. 2006], the communication included meta symbols of interaction between the robot and the human.

Online motion imitation by a humanoid robot

Recent studies focused on online robot motion generation based on captured data. First, we consider in [Dariush et al. 2008], where the authors developed a methodology to retarget human motion data to the humanoid robot ASIMO. Human motion was captured using a markerless pose tracking system. Here, upper body motion was considered by the Cartesian positions of the waist, wrists, elbows, shoulders and neck. The corresponding joint motion on the humanoid was calculated using inverse kinematics, while respecting the individual joint limits. In this case, a separate balance controller was used to move the legs in order to compensate for the retargeted motion of the upper body. In a later study, the same authors used a learning approach to pre-generate knowledge about human postures [Dariush et al. 2008]. During the actual motion retargeting, head and torso motion was monitored and a template closest from the ones learned was selected for further modifications. The arms were analyzed as 3D blobs and their position estimated. From this data the 3D features for head, shoulder, waist, elbows and wrists were localized. Using inverse kinematics and the balance controller the motion was computed, and then played on the humanoid robot in real-time.

Control Based Imitation

Yamane et al. [Yamane and Hodgins 2009], approached the online imitation problem by simultaneously tracking motion capture data and balancing the robot by a tracking controller and a balance controller. The balance controller was a linear quadratic regulator designed for an inverted pendulum model of the robot. The tracking controller computed joint torques to minimize the difference from the desired human capture data while considering full-body dynamics.

Whole Body Imitation

The authors in [Kim et al. 2009] studied the transfer of human motion including feet motion and dynamics. The dynamic behavior of the motion is given by an estimated trajectory of the ZMP. This trajectory is computed by solving an optimization problem that determines the inertial parameters of a simplified human model composed of solid boxes and spheres. The objective function is given by the squared distance between the ZMP of the actor, computed from measured reaction forces and torques, and the ZMP of the simplified model. Then, this ZMP trajectory is adapted to the robot by using the foot prints of the human. The upper body motion is solved by using an optimization problem to cope with size differences between the human and robot. The motion of the feet is computed from the footprints and the human refer-



Figure 1.13: Every year, LAAS-CNRS open the doors of the laboratory to showcase its research to the general public. The event generates a lot of interest especially about families with young children, HRP-2 is the hero of the day. In the 2009 event, our research group presented the HRP-2 platform including a demonstration of motion imitation. During one of the demonstrations, one child started imitating the robot balancing on one foot, who was in turn imitating a human actor doing the same. The core of this thesis presents the framework with which such imitation can be implemented. Such examples also show the powerful capability of human-like robots to attract and nurture scientific interest, especially in young students.

ence motion by using inverse kinematics. In order to achieve a performance of this method three controllers additionally are required: pelvis-ZMP controller, pelvis-orientation controller and foot-landing controller.

1.4 Problem Statement

Today's humanoid robots are capable of walking and balancing. A lot of effort is being done to make them more autonomous by incorporating the perception, planning and action loop. This loop forms an important basis in the research of human behavior while performing tasks. Perception refers to knowing what is happening in the environment, planning refers to making decisions on what motions the robot should perform to achieve specific goals in the environment, and action refers to the performance of planned motion to modify the environment or behave accordingly. One of the ultimate objectives for humanoid robots, as stated before, is to cohabit with humans. In this sense, robots require realtime reactivity, because of the unpredictable nature of human actions. This is a big challenge due to the complexity of the robot model.

In humans, imitation is an advanced behavior whereby an individual observes and replicates another human actions. Any action is defined as *imitation* if it resembles something that the actor has observed, the resemblance being for a long enough period of time such that it could not have occurred by chance [Kaye 1982]. It is accepted by scientists that children imitate adult behaviors from an early age [Jones 2006]. A simple example of this can be considered in the social settings of a family gathering. As one says 'good bye' to a baby by waving his/her hands, the baby reacts by doing the same. While this seems a rather trivial gesture, the implications are important. The baby mimics the hand waving motion by observation even if the motion itself carries little meaning to it. In neuroscience, researchers have suggested the existence of mirror neurons which form an important mechanism in such imitation and also in language acquisition. Mirror neurons were observed first by G. Rizzolatti and colleagues [Rizzolatti and Craighero 2004]. They noticed that some of the neurons in a monkey were activated from the fact that a monkey saw a person pick up a piece of food as well as when the monkey picked up the food itself. Another interesting anecdote of unintentional imitation is presented in Figure 1.13.

However, other kinds of motion are more complex to develop by imitating. Generally some reasoning and understanding about the motion itself is required to perform well. Examples of these are sports and dance. That is, these movements involve accurate performances to attain a predefined goal and often require training. In the context of robotics, we study the problem of online human motion imitation by humanoid robots. The main objective is that the humanoid observes human motion and replicates it to some extent. At the early stage of the motion imitation process in humans, the human brain identifies and represents salient features of the motion



Figure 1.14: Picture of a human performer extending a handshake and the humanoid robot HRP-2 imitating the gesture. The human motion was tracked using reflective motion markers and transferred in real-time to the humanoid.

being performed. In this sense, the first aspect we studied was the representation of the human motion aiming to realize online human motion transfer to a humanoid robot. The main difficulties to overcome, in developing human motion for humanoids, are the anthropomorphic differences between the human and the humanoid, the physical limits of the robot's actuators, the balance of the robot and the self-collisions. As a second point, we analyze these aspects to reliably transfer human motion to a humanoid robot online. We refer to the process of perceiving motion (via motion capture) and reproducing it (transfer from human to humanoid) as imitation. The platform for our experiments is the humanoid robot HRP-2 [Kaneko et al. 2004]. The Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS where this work was accomplished has the humanoid HRP-2 number 14 as one of its robotic platforms. HRP2 is depicted in Figure 1.14.

1.5 Contribution

The main topics on human motion imitation are transferring or retargeting captured data and learning human behaviors. In this Thesis we concentrate on online transfer of captured data to a humanoid robot HRP-2. We design a framework to transfer human motion to the humanoid based on the prioritization of task and inverse kinematics. The outcomes related to this work are,

- A *normalized representation* of human motion that is the vehicle for going from human motion to robot motion generation. This representation encodes motion properties like

positions, orientations and vectors, in contrast to the methods that directly use the joint trajectories. Furthermore, this representation allows to transfer motion to characters with different sizes and shapes. This representation is the result of fusion of the motion properties analysed in neuroscience research and computer animation area. From neuroscience the idea of using orientation normals to represent the motion properties was adopted, while the idea of attaching planes to upper limbs was developed further from the original ideas used in computer animation.

- An *anticipation model* for the center of mass position induced from feet positions and head motion. In the situation where the robot lifts a foot, it is very important to manipulate the center of mass of the robot properly in order to avoid instability. The model was inspired from neuroscience, from the hypothesis that feet and head influences the trajectory of the center of mass in humans.
- A framework for *online transfer* of human motions to humanoid robots. From a practical point of view, real-time imitation is a big challenge due to issues like the integration of continuous human motion data, motion representation, robot motion generation and communication between modules with the controller of the humanoid. Continuous motion data was guaranteed by using a linear kalman filter to predict the position of markers in case of glitches in the motion capture system. The normalized human representation and the motion generation based on task prioritization and inverse kinematics allows us to generate fast and stable motion for the humanoid robot.

1.6 Thesis Outline

In this chapter we reviewed the historical facts in motion measurement and robotics, as well as provide an overview of state of the art approaches. We then established the problems dealt with in this thesis and its final contributions.

In Chapter 2 we review the characteristics of HRP-2, and develop the main results on the approaches used to solve the inverse kinematics problem. We focus our attention on prioritized inverse kinematics method which is widely used in this work to generate robot motion.

The Humanoid-Normalized model is presented in Chapter 3. We review the principal representation of human motion and some methods used in computer animation. Furthermore, the motivation for the Humanoid Normalized model is discussed, and the validation of the proposed representation is presented.

The framework for transferring human upper-body motion with feet fixed to the ground is detailed in Chapter 4. We present some experimental results and discuss about the limitations of our approach. Also, practical issues like computational delays and software architecture are discussed.

In Chapter 5 we present the method where imitation while stepping can be implemented. Here, we go from upper body motions to whole body imitation including feet motion. The critical issues on timing and balance are also included.

Finally, in Chapter 6 we present the conclusions, remarks and sketch perspectives for further work.

1.7 Publication List

1. Francisco-Javier Montecillo-Puente, Manish N. Sreenivasa and Jean-Paul Laumond, On Real-Time Whole-Body Human to Humanoid Motion Transfer, International Conference on Informatics in Control, Automation and Robotics (ICINCO) 2010.
2. Francisco-Javier Montecillo-Puente et Jean-Paul Laumond, Imitation en ligne du mouvement humain par HRP2, Colloque JNHR 2010, Poitiers.
3. Francisco-Javier Montecillo-Puente et Jean-Paul Laumond, Imitation des mouvements humains par un robot humanoïde HRP2, CONGRES EDSYS, Toulouse 2009.

2

Inverse Kinematics and Tasks

In general, humanoid and manipulator robots are composed of a mechanical structure, actuators and sensors. The structure is built from rigid bodies, called links, which are connected by physical joints. These joints could be actuated, generally driven by electric motors or hydraulic devices. Particularly, humanoids are anthropomorphic robots whose overall appearance looks like the human body. The humanoid is modeled as a hierarchical structure of rigid bodies connected by revolute joints. In Figures 2.1 and 2.2 are shown a typical representation of a kinematic model of a humanoid and its hierarchy tree, respectively. Generally, the structure comprises head, torso, arms and legs.

In contrast with robot manipulators, the humanoid robot is not fixed to its work space. Hence, its motion is determined by its joints positions and a local frame attached to a specific body of the robot, called the root or base of the robot. The actuated joints allows humanoids to displace in the environment and manipulate objects simultaneously. Its displacement is achieved by the reaction forces created between contact points of the robot's feet and the ground. These reaction forces are also used to maintain its balance. The main advantage of a humanoid robot is its high mobility. That is, the mechanism is redundant with respect to manipulation or locomotion tasks which can be accomplished in different postures. In particular, for our experiments we used the humanoid robot HRP-2 number 14, which is available in our Laboratory. HRP-2 (Figure 2.3) is a humanoid robotics research platform, which is the result of the Japanese project Humanoid Robotic Project, HRP. This HRP project was run by the Ministry of Economy, Trade and Industry (METI) of Japan from 1998 to 2002 [Kaneko et al. 2004].

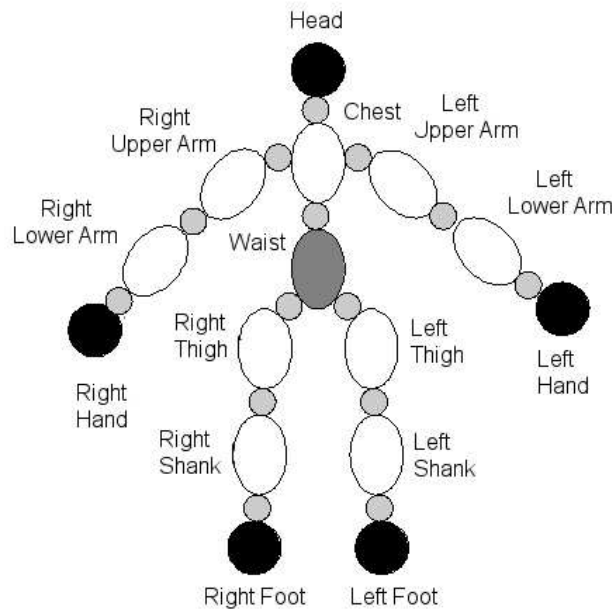


Figure 2.1: A rigid-body representation of a humanoid robot: the gray circles represents joints, the ellipses are for rigid bodies and the black circles indicate the terminal bodies(end-effectors).

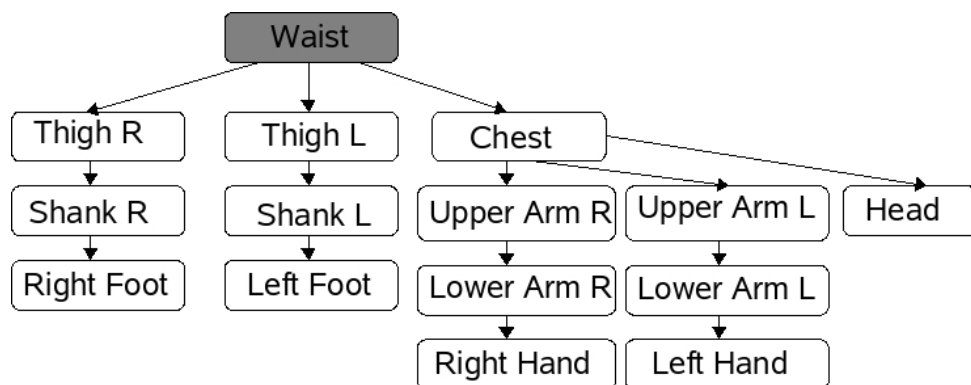


Figure 2.2: Hierarchical representation of the kinematic tree: rectangles represents rigid bodies, the arrows express a parent-child relation between bodies, R is for right and L is for left.

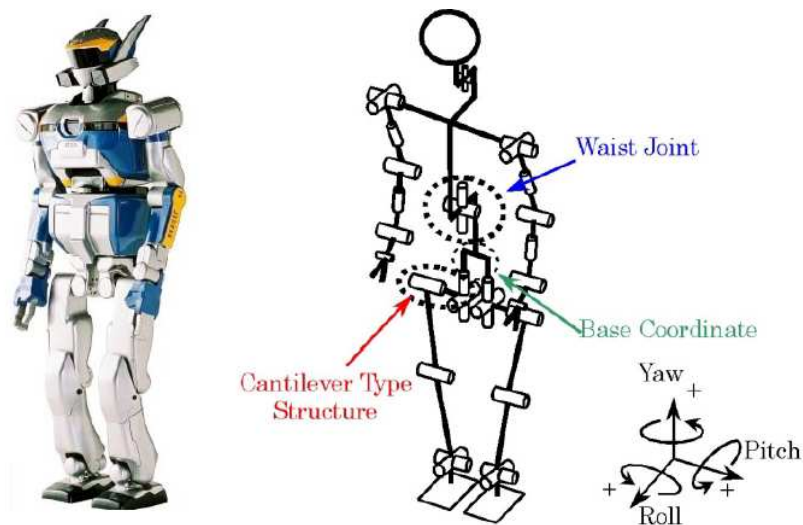


Figure 2.3: HRP-2 appearance and kinematic structure.

HRP-2 robot is constructed by Kawada Industries. It features 30 degrees of freedom, 56 Kg weight and 1.54 m height. Some of its other specifications are given in Figure 2.4. The range of motion for each joint is specified in Figure 2.5. In this table, the angles of a standard human are also given we can note that the range of motion are comparable. However, in the human there are more articulations that are not present in HRP-2, the spine for example features only 2 degrees of freedom. In the comparison of course the flexibility of the human body should also be remarked. From the hardware point of view, HRP-2 has two CPU in its body. One of them is utilized for the real-time controller of whole body motion, while the other is utilized for non-real-time control system including the vision and the sound systems.

In the following we review the current methods of Inverse Kinematics which has been used extensively to control robot manipulators. It has been also used to control more complex robots, like humanoid robots, and to animate digital characters.

2.1 Inverse Kinematics

Kinematics is a branch of classical mechanics concerned with the geometrically possible motion of a body or a system of bodies without considering the forces involved on generating the motion. In this prospect, the kinematic model of the system of bodies is the main information. Figure 2.6 represents a simple kinematic chain. In a robot manipulator, a tool is generally mounted on the terminal body and is called end-effector.

The geometric model of a manipulator gives the position and orientation of its end-effector. The set of all possible states of the end-effector is the work space of the manipulator and represents a subset of $\mathbb{R}^3 \times SO3$. The m joint parameters in the kinematic model span, on the other

Dimensions	Height	1,539 [mm]
	Width	621 [mm]
	Depth	355 [mm]
Weight inc. batteries		58 [kg]
D.O.F.	Total 30 D.O.F.	
	Head	2 D.O.F.
	Arm	2 Arms \times 6 D.O.F.
	Hand	2 Hands \times 1 D.O.F.
	Waist	2 D.O.F.
	Leg	2 Legs \times 6 D.O.F.
Walking Speed		up to 2.5 [km/h]

Figure 2.4: HRP-2 main specifications: number of degrees of freedom, weight, height, and maximum walking speed.

hand, a subset of \mathbb{R}^m , called the joint space.

Let $q \in \mathbb{R}^m$ be the configuration of the manipulator, the state $x \in \mathbb{R}^3 \times SO3$ of its end-effector is found through the evaluation of a non-linear function $f(q)$, which we write as:

$$x = f(q) \quad (2.1)$$

The inverse problem, consisting in finding a configuration satisfying a desired state of the end-effector is expressed through

$$q = f^{-1}(x) \quad (2.2)$$

The manipulator may be used to track a trajectory, rather than a single position. For example, when the manipulator is used for a painting task it is required to follow a predefined position and orientation path. This kind of tasks are represented by functions similar to the Equation 2.1.

The inverse kinematic designates the techniques that are used to solve this problem. This problem can be solved analytically, or numerically using the Jacobian matrix of the task [Liégeois 1977] [Nakamura 1991] [Siciliano and Slotine 1991] [Baerlocher and Boulic 1998] [Maciejewski and Klein 1985], or by nonlinear optimization [Zhao and Badler 1994].

The analytical techniques has the advantages of fast computation but limited to manipulators with small number of joints. Furthermore, the close-form solution depends on the physical structure of the manipulator. The solution based on the Jacobian matrix has the advantages of generality and treatment of multiple tasks at the same time. The methods based on optimization are more generic and have the advantage of incorporating equality and inequality constraints.

Joint			(a) Standard Human	(b) HRP-2
Head		R	-50 deg. to 50 deg.	no existence.
		P	-50 deg. to 60 deg.	-30 deg. to 45 deg.
		Y	-70 deg. to 70 deg.	-45 deg. to 45 deg.
Right Arm	Shoul- -der	R	-90 deg. to 0 deg.	-95 deg. to 10 deg.
		P	-180 deg. to 50 deg.	-180 deg. to 60 deg.
		Y	-90 deg. to 90 deg.	-90 deg. to 90 deg.
	Elbow	P	-145 deg. to 0 deg.	-135 deg. to 0 deg.
		Y	-90 deg. to 90 deg.	-90 deg. to 90 deg.
	Wrist	R	-55 deg. to 25 deg.	no existence
		P	-70 deg. to 90 deg.	-90 deg. to 90 deg.
Right Hand		P	0 deg. to 90 deg.	-16 deg. to 60 deg.
Waist		R	-50 deg. to 50 deg.	no existence
		P	-30 deg. to 45 deg.	-5 deg. to 60 deg.
		Y	-40 deg. to 40 deg.	-45 deg. to 45 deg.
Right Leg	Hip	R	-45 deg. to 20 deg.	-35 deg. to 20 deg.
		P	-125 deg. to 15 deg.	-125 deg. to 42 deg.
		Y	-45 deg. to 45 deg.	-45 deg. to 30 deg.
	Knee	P	-0 deg. to 130 deg.	-0 deg. to 150 deg.
	Ankle	R	-20 deg. to 30 deg.	-20 deg. to 35 deg.
		P	-20 deg. to 45 deg.	-75 deg. to 42 deg.

R: Roll axis, P: Pitch axis, Y: Yaw axis

Figure 2.5: Range of motion for the articulations of standard Japanese people and the corresponding range of motion for each joint in HRP-2, [Kaneko et al. 2004].

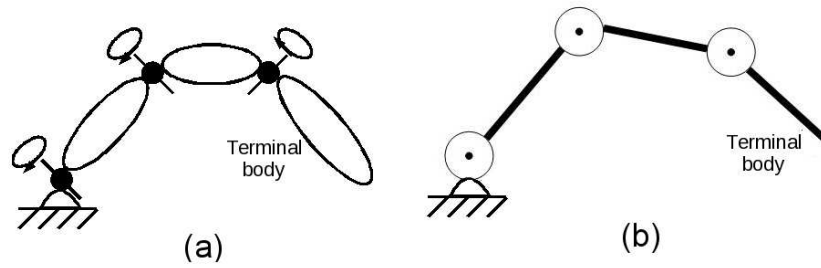


Figure 2.6: (a) A simple manipulator and (b) its geometric model.

The inequality constraints arise in several situations, for example in collision avoidance tasks and joint limits. However, they are much more costly.

In order to solve the inverse kinematic problem by the Jacobian pseudo inverse, the differential of the Equation 2.1 is used. This differential equation gives us the relationship between the variation of the joint parameters δq and the corresponding displacement δx of the position and orientation of the terminal body (or end-effector), that is,

$$\delta x = J(q)\delta q \quad (2.3)$$

note that this equation is a first order approximation of Equation 2.1, where

$$J(q) = \frac{\delta f(q)}{\delta q} \quad (2.4)$$

is the $m \times n$ Jacobian matrix of the function $f(\cdot)$ in Equation 2.1, m is the dimension of the task, and n is the number of degrees of freedom. Henceforth, we use J to express $J(q)$.

The solution of Equation 2.3 in terms of δq determines the joint variation that produces an expected displacement δx . If $m = n$ and J is non singular the joint variation is simply

$$\delta q = J^{-1}\delta x \quad (2.5)$$

where J^{-1} is the inverse of J . If the dimension of the task is greater than the number of degrees of freedom, $m > n$, Equation 2.3 may not have solutions.

If the dimension of the task is lower than the number of degrees of freedom, $m < n$, and J is non singular, there exists several solutions of δq . The general least-squared solutions for this case are given by

$$\delta q = J^+\delta x + (I_n - J^+J)z \quad (2.6)$$

where $J^+ = J^T(JJ^T)^{-1}$ is the pseudo inverse of J , I_n is the $n \times n$ identity matrix, and z is an

n -dimensional arbitrary vector. These solutions minimize the norm $\frac{1}{2} \|\delta x - J\delta q\|_2^2$. The first term on the right side of Equation 2.6 is the minimum norm solution, the second term is used to reach all the possible solutions. This is referred as the orthogonal projector on the null space of J .

The second term on the right side of Equation 2.6 could be exploited to solve a secondary task as reported in [Liégeois 1977].

2.2 Inverse Kinematics: Solving Two Tasks

Several authors had investigated the problem of solving several tasks in the inverse kinematic framework [Liégeois 1977] [Nakamura 1991] [Siciliano and Slotine 1991] [Baerlocher and Boulic 1998] [Maciejewski and Klein 1985]. The common problem to solve in these works was to formalize the conflict between tasks. The main two approaches to solve the inverse kinematic problem with multiple tasks are based on defining priority on each task or in defining a single task that is composed from the weighted sum of several tasks. We selected to use the solution based on priority because it guarantees that some tasks are solved more accurately than others. For example, in Figure 2.7 there is a robot with two end-effectors. We define two tasks, one for the end-effector A to reach the point P_A and other for the end-effector B to reach the point P_B . In this example, if the end-effector A reaches the point P_A , the end-effector B does not achieve its task, and the other way around. A general solution to this problem is to establish a predefined priority for each task. For example, first solve the problem for one task and then use its null-space projector to solve the second task. This schema of prioritization is useful for humanoid balance, i.e. we first solve a task to control its center of mass, closely related to the balance problem.

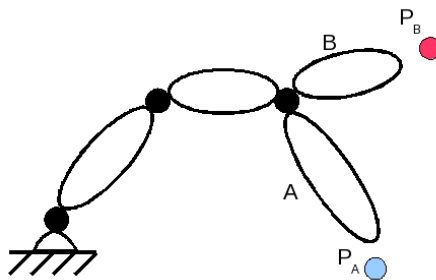


Figure 2.7: Conflicting tasks, if end-effector A reaches the location P_A then P_B cannot be reached by the end-effector B.

For example, if we have two tasks $x_1 = f_1(q)$ and $x_2 = f_2(q)$, where x_1 and x_2 are the first and second priority tasks for the robot, respectively. We could determine a solution according

to this order of priority. First, the variation of q that solves the task according to the priority is determined from the differential kinematics. The differential for both tasks are given by,

$$\delta x_1 = J_1 \delta q \quad (2.7)$$

$$\delta x_2 = J_2 \delta q \quad (2.8)$$

The task x_1 has an infinity of solutions if the task dimension is $m < n$ and J_1 is not singular. Its general solution is given by Equation 2.6, that is,

$$\delta q = J_1^+ \delta x_1 + (I_n - J_1^+ J_1) z_1 \quad (2.9)$$

where J_1^+ is the pseudo inverse of J_1 , I_n is the $n \times n$ identity matrix and z_1 is an arbitrary n -dimensional vector.

Substituting Equation 2.9 into Equation 2.8, we have

$$J_2(I_n - J_1^+ J_1) z_1 = \delta x_2 - J_2 J_1^+ \delta x_1 \quad (2.10)$$

Then solving Equation 2.10 for z_1 , in the same way as Equation 2.3, we obtain

$$z_1 = \hat{J}_2^+ (\delta x_2 - J_2 J_1^+ \delta x_1) + (I_n - \hat{J}_2^+ \hat{J}_2) z_2 \quad (2.11)$$

where $\hat{J}_2 = J_2(I_n - J_1^+ J_1)$ and z_2 is an arbitrary n -dimensional vector.

Finally, the solution of two tasks with priority is solved from Equations 2.10 and 2.9. After reducing some terms we have,

$$\delta q = J_1^+ \delta x_1 + \hat{J}_2^+ (\delta x_2 - J_2 J_1^+ \delta x_1) + (I_n - J_1^+ J_1)(I_n - \hat{J}_2^+ \hat{J}_2) z_2 \quad (2.12)$$

2.3 Inverse Kinematics: Solving Multiple Tasks with Priority

The Least-Squares solutions for two task was extended for multi-chain robots, Figure 2.8. In this case, the vector z_2 in Equation 2.12 is exploited now to solve a third task. The procedure to find the solution for the third task is similar as solving for two prioritized tasks. Furthermore, this procedure can be generalized to solve n prioritized tasks, $x_i = f_i(q)$ with $i = 1 \dots n$ in descending order of priority.

The general solution for n tasks could be expressed by the following iterative expressions,

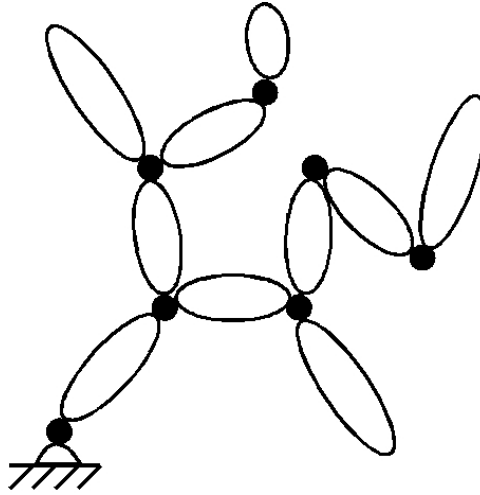


Figure 2.8: A robot manipulator with multiple kinematic chains.

$$N_0 = I_n \quad (2.13)$$

$$\hat{J}_i = J_i N_{i-1} \quad (2.14)$$

$$N_i = I_n - \hat{J}_i^+ \hat{J}_i \quad (2.15)$$

$$\delta q_{i+1} = \hat{J}_{i+1}^+ (\delta x_{i+1} - J_{i+1} \delta q_i) \quad (2.16)$$

where N_i is the projector associated with the i^{th} task, J_i is the Jacobian matrix of the i^{th} task, I is the $n \times n$ identity matrix.

2.4 Damped Inverse Kinematics

A point q is called singular if $\det J(q) = 0$, in the case where $m = n$ in Equation 2.3. For a redundant robot manipulator $m < n$, the point q where the Jacobian matrix has not full rank is called a singular point.

In practice, for the least-squares solutions some problems are found around the manipulator singularities. Near singularity Equation 2.6 can produce large values of δq because of the ill conditioning of the Jacobian matrix. Wampler and Nakamura proposed to use a damped least squared solution and singular robust inverse to overcome the ill conditioning of the Jacobian matrix, respectively [Wampler 1986] [Nakamura and Hanafusa 1986]. For a single task, the problem to be solved was changed from the least-squares solution to

$$\min_{\delta q} \|\delta x - J\delta q\|^2 + \alpha^2 \|q\|^2 \quad (2.17)$$

where $\alpha \in \mathbb{R}^+$ is a damping factor. The solution to this problem is given by

$$\delta q = J^* \delta x \quad (2.18)$$

$$J = J^T (JJ^T + \alpha^2 I_n)^{-1} \quad (2.19)$$

We can observe that if $\alpha = 0$ we have the definition of the classical pseudo-inverse matrix J^+ . All the damped least squared solutions can be obtained by replacing J^+ by J in Equation 2.6, that is

$$\delta q_\alpha = J^* \delta x + (I_n - J^* J) z \quad (2.20)$$

The good conditioning of J^* and continuity of Equation 2.20, can be analyzed using the Singular Value Decomposition (SVD). The SVD of J^* is given by

$$J^* = U \Sigma V^T = \sum_{i=1}^m \frac{\sigma_i}{\sigma_i^2 + \alpha^2} v_i u_i^T \quad (2.21)$$

where U is an orthonormal matrix $m \times m$ of the output singular vectors u_i , V is a $n \times n$ orthogonal matrix of input singular vectors v_i , and $\Sigma = [D \ 0_n]$ is a $m \times n$ matrix whose $m \times m$ diagonal submatrix D contains the eigenvalues σ_i of the matrix J^* and 0 is a $m \times (n - m)$ zero submatrix.

Substituting J^* for its SVD decomposition in Equation 2.20, we have

$$\delta q_\alpha = \sum_{i=1}^m \frac{\sigma_i (u_i^T \delta x) + \alpha^2 (v_i^T \delta z)}{\sigma_i^2 + \alpha^2} v_i + \sum_{i=m+1}^n (v_i^T \delta z) v_i \quad (2.22)$$

As we can observe in Equation 2.22, that near a singularity (when σ_i is close to zero) and with a non zero damping factor the good conditioning and continuity are guaranteed. However, the accuracy of the solutions is decreased. This approach could also applies to multiple prioritized tasks.

2.5 Nonlinear Programming: Inverse Kinematics, Tasks and Constraints

The inverse kinematics methods based on the pseudo inverse of the Jacobian and the damped least squared have the disadvantage that it is not guaranteed completion of tasks with zero errors because of the priority order. What is more inequality task cannot be added directly.

As mentioned previously, the inverse kinematics could be solved by using nonlinear programming [Zhao and Badler 1994]. The general inverse kinematic problem is stated now as

$$\begin{aligned}
& \min_{\delta q} && F(\delta q) \\
& \text{s.t.} && \\
& && H\delta q + a = 0 \\
& && G\delta q + b \leq 0
\end{aligned} \tag{2.23}$$

The objective function $F(\cdot)$ is used to solve multiple tasks or to define a meaningful quantity to be minimized. For example, we minimize $F(\delta q) = \sum_i w_i \|J\delta q - \delta x_i\|^2$, where w_i is a weighting factor for each task x_i . A feasible solution is determined for this problem if all equality and inequality are satisfied. However, special attention should be made for the objective function because it is only minimized. For example, if we have a task that must be strictly satisfied instead of adding this task to $F(\cdot)$, we can add it as an equality constraint. The main drawback of this formulation can be a larger computation time.

2.6 Framework for Control of Redundant Manipulators

To control manipulators via the Jacobian matrix the resolved motion control was proposed in [Whitney 1969]. Figure 2.9 shows the basic outline for this kind of control. In the original formulation the inverse of the Jacobian was used, in our case the pseudo inverse of the Jacobian matrix is used instead to control redundant robots. The robot updated its configuration from the solution of the inverse kinematic problem. This schema of control could be extended to multiple tasks with priority or even the inverse kinematic problem could be solved using optimization methods.

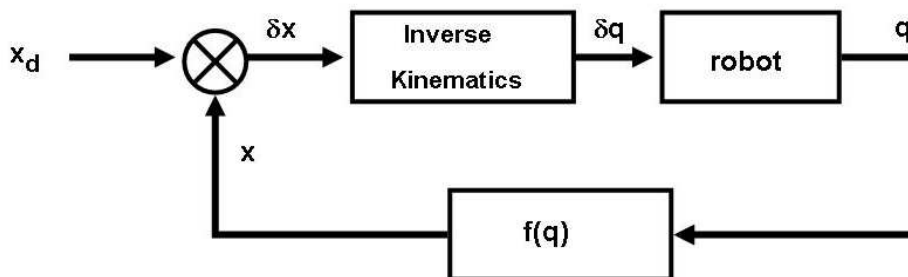


Figure 2.9: Basic outline to control redundant manipulators using Jacobian matrix.

Recalling the inverse kinematic solution we have the following relations

$$\begin{aligned}\delta x &= J(q)\delta q \\ \delta q &= J^+ \delta x + (I_n - J^+ J)z\end{aligned}$$

In this schema of control, the expressions involved are given by,

$$x = f(q) \tag{2.24}$$

$$\delta x = x_d - x \tag{2.25}$$

$$q' = q + \delta q \tag{2.26}$$

$$\tag{2.27}$$

where x is the current state of the robot, x_d is the desired state, δq is the increase on the current configuration in order to attain x_d and q' is the updated configuration of the robot. Note that this schema is useful only for the case where δx is small, i.e. for moderated speeds.

2.7 Conclusions

In this Chapter we reviewed a kinematic framework for control of redundant manipulators and developed the main tools used for our work. In particular, we described the humanoid robot HRP-2 which is the platform used for our experiments. The main results concerning to prioritized inverse kinematic were overviewed. We have studied the problem of solving multiple tasks in a prioritized descending order. The most interesting aspects of the prioritized inverse kinematics based on pseudo inverse Jacobian are its generality and fast computation in practice. Finally, the solution of the inverse kinematic problem using optimization techniques was also discussed.

3

Human Motion Representation

In this Chapter we study the representations of human motion and propose a Humanoid Normalized model used as vehicle to transfer human motion towards a humanoid robot. The human motion is recorded using a motion capture system.

Motion capture is the process of recording motion data. This is usually done by tracking reflective velcro markers attached to the body of a subject using infrared cameras. Other systems to capture human motion are based mechanic or electromagnetic devices. This process has been widely used for entertainment, sports and medical applications.

In some animated films, the captured human motion has been used to animate digital characters. Usually the captured data and the target character have differences in size and anthropometry. This problem is solved by a process called “Retargeting” [Gleicher 1998]. In robotics, motion capture has been used to generate dance motion on humanoid robots [Nakaoka et al. 2003].

Human motion representation is not unique, and depending on the application several representations have been adopted. One choice is represent to it by a set of identified markers connected by links, external markers representation. Figure 3.1 shows a picture of a marker set and its links. In this representation only information about marker positions, body parts marker attachments, and links between markers is available.

There is another representation based on positions, but instead of only using marker positions, estimated centers of human body articulations are used instead, see Figure 3.2. Additionally to these centers a frame, called root frame, is also recorded. This representation is the

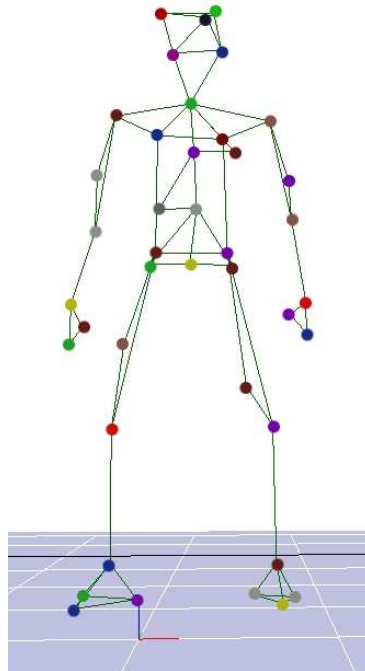


Figure 3.1: Human motion representation by markers positions and links, external markers representation.

result of an specific marker set of the external marker representation to recover the centers of the joints.



Figure 3.2: Human motion representation by joint centers: adapted from [O'Brien et al. 2000], the motion capture system used in this case was based on magnetic devices.

Angles trajectories (skeleton representation) is the representation more frequently used in animation, see Figure 3.3. In this representation a skeleton is created first, then it is animated based on sensed (or computed) joint angles. The most common devices for sensing or determine these angles are based on reflective markers and infrared cameras, inertial sensors, mechanical

devices and even magnetic systems.

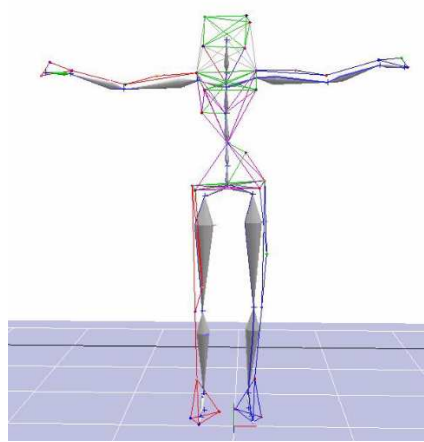


Figure 3.3: Human motion represented by joint trajectories of an skeleton.

There are others representations which are application specific. For instance the Labanotation, which is a symbolic methodology to represent dance motions. It was proposed by Rudolf Laban in 1928. This notation expresses positional information of body parts at distinct instants of time. It is read down to top and divided into columns that group symbols. These symbols codifies information about the body parts they affect. The columns are arranged beside a center line to represent left-right symmetry of the body. Some work has been proposed to read Labanotation charts and to reproduce the codified dances [Yu et al. 2005] [Wilke et al. 2005].

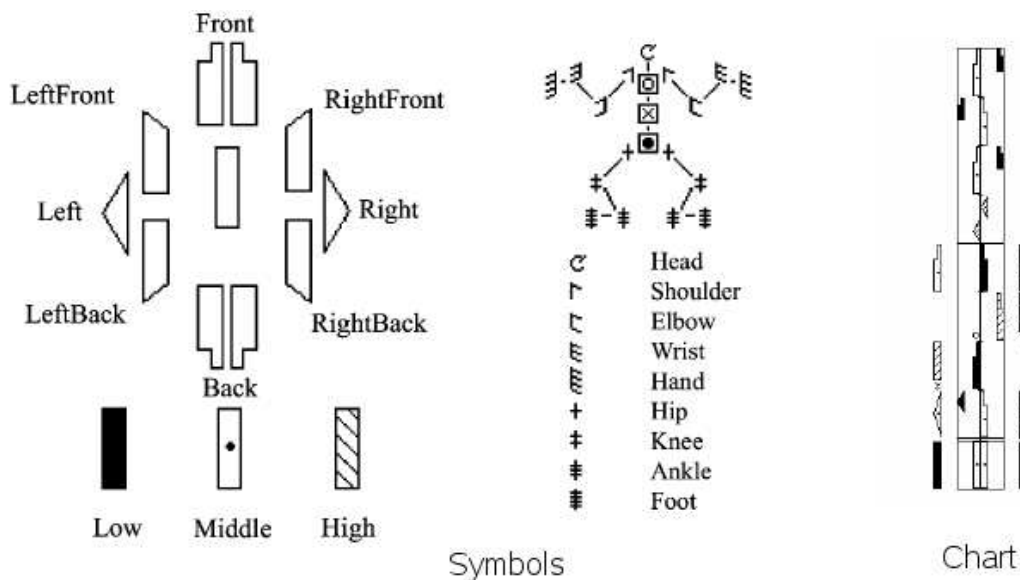


Figure 3.4: The symbols used in Labanotation and a chart representing dance motions.

In the work developed by Kulic and Nakamura [Kulic and Nakamura 2009], the authors

studied the influence of the motion representation in motion learning and segmentation. This work includes comparisons between the skeleton representation using quaternions, Euler angles and the Cartesian position representation (joint centers). The main result reported is that the Cartesian positions performed better for motion segmentation than the others.

3.1 Humanoid Normalized Model

An skeleton is a set of bodies connected with revolute joints. The usual form to represent the motion using an skeleton is to record the angle trajectories for each joint. From these trajectories and the parameters of the skeleton (length of bodies and revolute joints) we can reconstruct the captured motion. For realtime analysis, this representation is not convenient because it is skeleton dependent and takes more computing time to adapt it to other skeletons. Moreover, this representation is too sensible if a marker is not identified for the system, that is, link orientation may be lost. Furthermore, we are interesting in transferring motion properties instead of unmeaning joint trajectories.

The information we chose to keep and represent was mainly inspired from the neuroscience research and the computer animation community.

Neuroscience research groups have reported how head, arm and trunk are coordinated in reaching tasks [Svestrup et al. 2008], and between head and trunk when humans walk [Patla et al. 1999]. These works show evidence of how humans coordinate their limbs to move.

The relevance of these works is about the information and the properties in the motion itself important to preserve or to study. In particular, we observed in these studies that the main source of information was marker position relations, rather than angles trajectories. These marker relations were defined to extract relevant information. For example, Figure 3.5 show a classical set of markers used in neuroscience for the study of the relations between different body parts. In this case, the authors are interested in the study of the relation between head and trunk orientations when walking. The head is considered as a rigid body, to recover its orientation three markers are placed on the face of the human, similar for the trunk. Only two markers are used to determine the feet position trajectory. The placement of the markers is generally determined empirically by testing several combinations.

From the computer animation point of view, retargeting is the process to transfer motion from one skeleton to another with anthropomorphic differences [Gleicher 1998]. In [Multon et al. 2008], the authors proposed an intermediate normalized entity to retarget motion to digital characters. This entity includes information on only some key joints of the skeleton, while other joints are computed at the retargeting time according to the target character, i.e. the intermediate joints for the arms and legs. The main idea for not saving the joints for arms and legs is to assume that these lie on a plane formed by their joints, i.e. the arm lies in a plane formed by

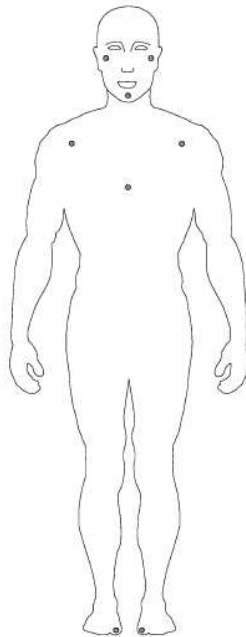


Figure 3.5: A marker set used in neuroscience adopted from [Patla et al. 1999]. Head and trunk were modeled as rigid bodies, hence three markers were used to recover their motion. To measure stride information one reflective marker was attached to each foot. In this work the authors studied the coordination process between head, trunk and center of mass.

the shoulder, elbow and wrist joints. These planes codifies in some way the posture of the arms and legs. These planes are shown in the Figure 3.6.

We name Humanoid Normalized model, henceforth HN model, to the representation of motion which codifies the motion by salient features of the whole body motion. This representation includes the idea of planes to represent the arm postures, and incorporates information about the hands, chest, head, waist, and feet, refer to Figure 3.7. The main purpose of building this model is to be used for transferring human motion to an humanoid robot. The selection of the features has been chosen because these quantities are to be preserved when transferring human motion to characters and has been of interest in neuroscience, computer animation and robotics studies.

Human motion was captured using a motion capture system equipped with 10 infra-red tracking cameras (MotionAnalysis, CA, USA). The system is capable of tracking the position of markers within a 5x5 m space within an accuracy of 1mm, at a data rate of 100 Hz. The motion was tracked from a human wearing 41 reflective markers firmly attached to its body using velcro straps, or tape (see Fig. 3.8), refer to the appendix A to observe in more detail the marker set used. From these markers, we use only 21 markers to build our HN model.

During each motion capture session the human starts form a standard posture: stand up with arms lying down and head looking forward. First, we rotate and translate all markers to a

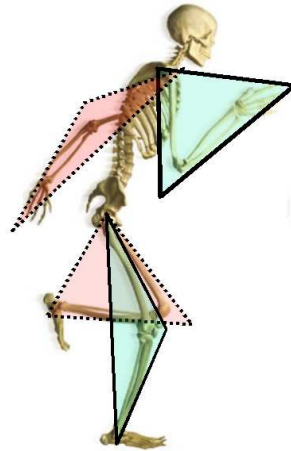


Figure 3.6: In MKM animation software it is assumed that legs and arms lies on planes.

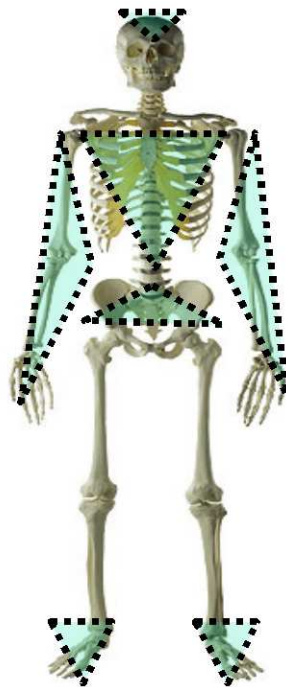


Figure 3.7: Planes in our HN model, we include planes for the chest, waist, head and feet.

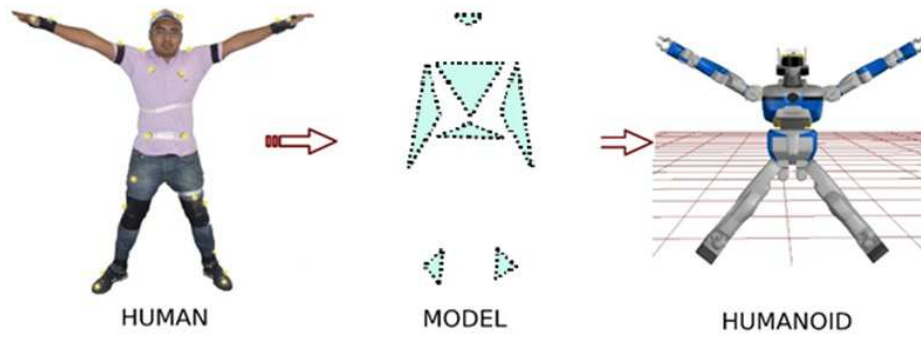


Figure 3.8: Human→Humanoid Normalized Model→Humanoid. Motion capture position data from the human is transferred on the normalized model and associated with the planes and humanoid joints. The motion of these planes and joints drives the humanoid motion.

predefined frame located on the floor based on this standard posture. The rotation is computed by using the feet markers, a rotation matrix is computed in order the x-axis is the forward direction and z-axis is up direction, and feet lies on the y-axis. The translation is computed in such a way that the origin of this predefined frame is located in the middle point between feet positions. The marker positions are relative to this origin, or to a fixed foot position.

Arm Motion The arm motion includes its posture and hand position. The main assumption we made as in [Multon et al. 2008] is that the arm lies in a plane. This virtual plane of the arm is formed by shoulder, elbow and wrist articulations. The normal to this plane and the position of the hand represent the arm motion. We use the markers attached to the shoulder, elbow and wrist to compute the plane normal and the hand markers to compute the hand position. The markers, the plane and its normal are shown in Figure 3.9.

The plane normal is computed by

$$N_{arm} = V_0 \times V_1 \quad (3.1)$$

where,

$$V_0 = \frac{p_0 - p_1}{\|p_0 - p_1\|}$$

$$V_1 = \frac{p_0 - p_2}{\|p_0 - p_2\|}$$

and p_0 , p_1 and p_2 are the marker position attached to the shoulder, elbow and wrist for either right or left arm.

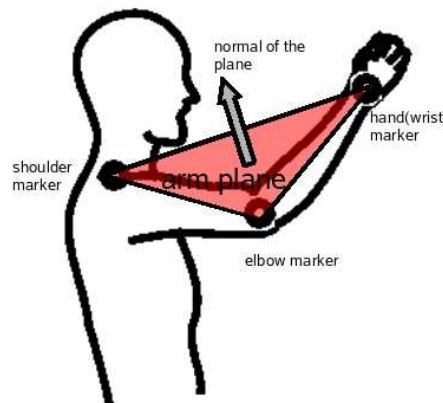


Figure 3.9: Right arm markers and its plane.

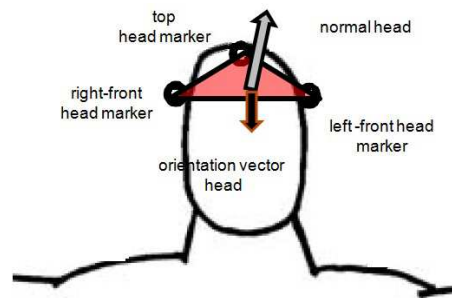


Figure 3.10: Head markers and orientation vector.

Head Motion The head motion is split into two components, its orientation and its position. The head is considered as a rigid body, so that only three markers are required to extract its orientation. Its position is computed from these markers. In the standard position of the head, the normal of the plane formed the attached markers is pointed upwards, figure 3.10. However, we use the vector perpendicular to this laying on the sagittal plane. Using only one vector we lost one degree of freedom on the head orientation. This is not a problem because in humans beings the roll rotation (rotation about the x-axis) is quite limited, so that it can be neglected. With the orientation vector we use the roll rotation is lost. The normal of the plane head is computed as for the arm, but head markers are used instead.

Chest Motion We represent the chest (or better trunk) as a rigid body, so that its orientation can be computed from three markers. A plane is also attached to the chest, which is formed from the front-left chest, front-right chest and low-mid back markers, Figure 3.11. The chest motion is represented by the normal vector to this plane. In this case, we lost one degree of freedom on the orientation, the roll rotation.

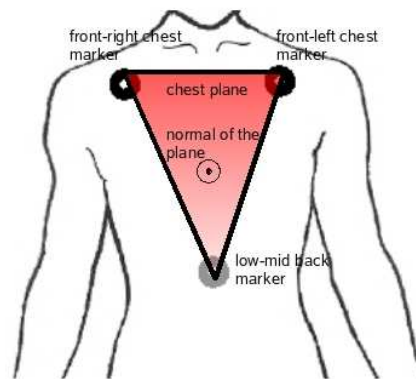


Figure 3.11: Chest markers and its plane.

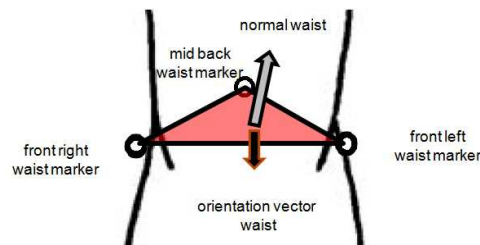


Figure 3.12: Waist markers and its plane.

Waist Motion Again, the best way to represent the waist motion is through a orientation vector. The markers right asis, left asis, and root are used to compute a normal vector, to locate these markers refer to the Appendix A. But, we represent the waist motion by a orientation vector, Figure 3.12. This is pointed in the forward direction and lies on the transversal plane. In this case, we lost one degree of freedom on the orientation, the roll rotation. The waist motion is not driven itself completely, but from the legs posture. That is, in the case of humanoid robots the balance is more important so that leg motion is used for this purpose. This fact constraints the range of motion of the waist.

Feet Motion The feet motion represented by its position and its yaw orientation. As for the head, we consider the feet as rigid bodies. We attach four markers named toe, hell, lateral ankle and medial ankle, Figure 3.13. From these, we compute its position in a similar way as for the head but the orientation vector is projected to the ground to keep only the yaw orientation component.

In practice, it is very important the placement of the markers, these must be located precisely at the anatomical points indicated in Appendix A. Particularly for our model, we must take care on the markers related to the arms: the shoulder, elbow and wrist markers. These must be placed

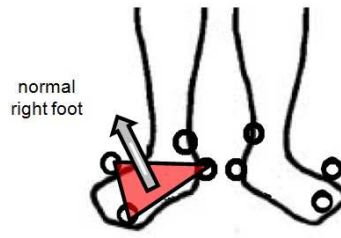


Figure 3.13: Right foot markers and its plane.

in way that are not in the same line when the arms are completely extended, no plane is defined there.

3.2 Humanoid Normalized Model Representation

The motion that we consider for each limb could be represented by positions P_{body} , normal N_{body} and orientation vectors V_{body} only. We put them together in an vector of features to represent the human motion,

$$[P_h, V_h, N_c, V_w, P_{lh}, N_{la}, P_{rh}, N_{ra}, P_{lf}, P_{rf}] \quad (3.2)$$

where P_h is a point representing the position of the head, V_h is a vector representing the orientation of the head, N_c is a vector representing the normal of the chest plane, V_w is a vector representing the orientation of the waist, P_{lh} is a point representing the position of the left hand, N_{la} is a vector representing the normal to the left arm plane, P_{rh} is a point representing the position of the right hand, N_{ra} is a vector representing the normal to the right arm plane, P_{lf} is a point representing the position of the left foot and P_{rf} is a point representing the position of the right foot. Additionally, in the case of humanoid robots we include, the *CoM* property. It is a point representing the center of mass projection to the ground. We will explain it in more detail in Chapter 4. Of course, this is a time varying model. In Fig. 3.14 a geometric representation of the HN model is shown.

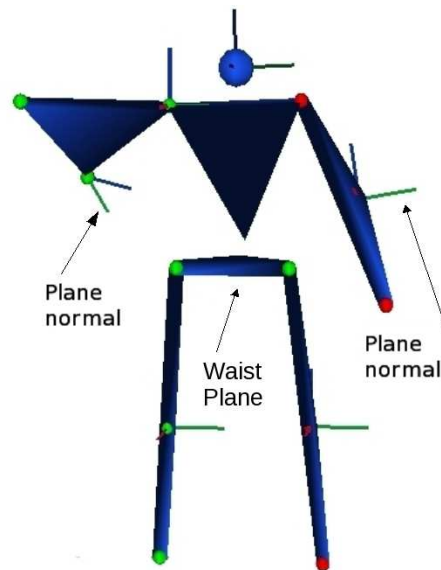


Figure 3.14: The geometric representation of the HN model. We observe the planes for the arms, chest and less clear the one for the waist. We also included a frame for each arms. The Y-axis represents the normal of the plane.

3.3 Model Validation

3.3.1 Human Motion Data: Skeleton and Markers

To validate the proposed representation of human motion we used a reference skeleton and reference joint angles trajectories. The left skeleton shown in Figure 3.15 is the reference skeleton. Some snapshots of the reference motion, called dance motion, are depicted in Figure 3.17. Provided that our motion capture system is capable to attach an skeleton to a set of markers and to compute its joint trajectories, we performed a motion capture session and recorded both the marker positions and the joint trajectories.

We attached 41 markers to the body of a subject test, then we built an skeleton. This skeleton was composed of 19 segments, it accounted with 32 degrees of freedom (dof), and the 6 dof's associated to the root reference frame, refer to Table 3.1. We considered 7 dof's for the legs (3 dofs's for hips, 1 dof for the knee and 3 dof's for the ankle), 2 dof's for the chest, 7 dof's for the arms (3 dof's for the shoulder, 1 dof for the elbow and 3 dof's for the wrist) and 2 dof's for the head. The parameters required to build a kinematic chain are provided by the motion capture system. Particularly, segments lengths, the type of joint connecting adjacent segments, and the number of degrees of freedom for each joint, refer to Figure 3.15.

Limb segment	Dofs
Head	2
Legs	$7 \times 2 = 14$
Arms	$7 \times 2 = 14$
Chest	2

Table 3.1: Distribution of the degrees of freedom in our reference skeleton.

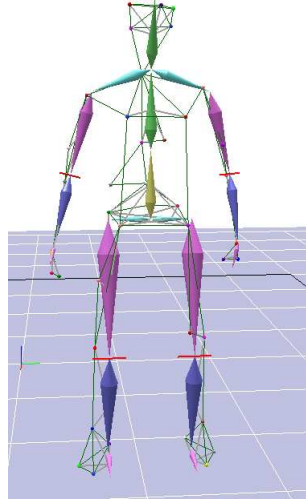


Figure 3.15: Reference skeleton used to validate our model, it is composed of 19 segments. From these, 4 are fixed the segments colored in cyan.

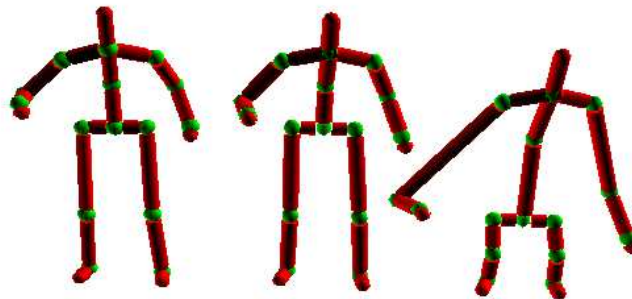


Figure 3.16: Models used to validate our model. The character at the left of the picture is the original skeleton, the character of the middle is a copy of the original character(kinematic structure adapted to be animated by our approach), the character at the right has the same kinematic structure than the original but differs in the segment lengths.

3.3.2 Skeleton Animated by Humanoid Normalized Model

We used the parameters of reference skeleton to build its kinematic chain. To use the prioritized inverse kinematic approach, we defined a stack of tasks to animate our target skeletons, based on the recorded marker positions and our HN representation for human motion. The tasks we used are the followings

1. Homogeneous transformation task for each feet, i.e. both position and orientation are fixed,
2. Position task for the head,
3. Homogeneous transformation task for the left wrist,
4. Homogeneous transformation task for the right wrist,
5. Orientation vector task for the chest,
6. Orientation vector task for the waist,
7. Orientation vector task for the head.

The order of the priority if these task where defined according to the its influence on the posture of the target character. First, feet tasks are defined to locate the character in the environment and to guarantee contacts between feet and ground. Second, we have observed that the position of the head influences the general posture of the robot, i.e. if it is moving down the character becomes in a half-sitting posture or the body leans towards the head target position. We can remark that the orientation head has been split until the final, that is because the dof's that define the orientation vector are available on the robot. Moreover, if higher priority was set on the head orientation task, the inverse kinematic solver uses more dof's to reach the orientation target which produces unreal postures.

The target entries for the tasks were computed from the HN model. The joint angles trajectories for the target character were computed from the solution of the stack of tasks by prioritized inverse kinematics.

To evaluate the quality of the motion transfer the aspects to consider are the visual similarity (qualitative) and the numeric comparisons (quantitative). Both are quite difficult to evaluate, the visual similarity depends on personal criteria. Moreover, the differences between kinematics models, size and structure produces incompatibilities between motions.

We evaluate the quality of the transfer by observing the motion at the moment of being generated, by computing the RMS errors of the joint trajectories and by computing the RMS errors of the motion properties in the representation we adopted.

First, some snapshots of the reference character motion and the computed target character motion are depicted in Figures 3.17 and 3.18. Also, Figure 3.19 shows a sequence of the motion transferred to a character with differences in size and limb lengths.

Segment	joint number or dof name	RMS
Waist frame	x position	0.024 m
	y position	0.037 m
	z position	0.014 m
	roll angle	1.530 deg
	pitch angle	3.950 deg
	yaw angle	3.670 deg
Left arm	joint 0	14.630 deg
	joint 1	23.900 deg
	joint 2	11.010 deg
	joint 3	22.790 deg
	joint 4	18.590 deg

Table 3.2: RMS of the waist frame (passive) and the left arm joints. The joint 0, joint 1 and joint 2 corresponds to the shoulder dof's in the order z-axis, y-axis, x-axis. The joint 3 corresponds to the elbow rotation, y-axis and the joint 4 corresponds to z-axis rotation. The convention is z-axis upwards, x-axis forward direction.

The quantitative aspect is evaluated by the RMS errors between the original motion and the motion of target character. In particular, we compare the RMS of the free frame (frame attached to the waist) and the joint angles of the left arm. The results are shown in the Table 3.2. These are of interest because the dof's of the free frame are passive (position and orientation of the waist), that is there are not driven by actuators but from the leg motion while the dof's of the arm are active. The error values are high, however when judging the visual aspect it is difficult to give an overall judge of the quality of the motion. This high values are not surprising we are not searching in preserving the joint trajectories, but the motion properties. To transfer these properties the redundant nature of the character is exploited, and the problems related to the range of motion and retargeting are solved at the same time. In the inverse kinematic solver the limits on the motion for each joint is included, and the retargeting is solved previously by the scaling stage and motion properties.

In the same sense, the RMS errors between the motion properties in our representation are resumed in the Table 3.3. The RMS errors are the magnitude of the difference vector between the reference normals and the normals of the target character. To say, the normal of the arms corresponds to the axis of rotation of the elbow joint, the normal of the chest is the vector axis of its local frame pointing in the forward direction. We interpret these quantities in the following way, if the RMS error were zero it means the 100% of the property was transferred. So the RMS error of the left arm normal means that $(1 - 0.103) \times 100\% = 89.7\%$ was transferred to the character. In this analysis, the importance of adopting a good representation to generate and to evaluate the motion is revealed by the RMS errors of the joint trajectories and the errors of properties in our representation, even if numerically cannot be directly compared. The error

Property	RMS (adimensional)
Left arm normal	0.103
Right arm normal	0.136
Chest normal	0.110
Head normal	0.284

Table 3.3: RMS errors of the motion properties in our representation.

in our representation give us an easy interpretation about the quality of the properties of the motion. However, for the joint trajectory errors is more difficult. In fact, this particular motion can be evaluated more easily from a qualitative point of view when the joint trajectories are played on its kinematic model.

Finally, Figure 3.20 shows plots of the first three joint of the left arm, the black solid lines represent the reference trajectories and the dashed red lines are the corresponding trajectories of the target character. Figures 3.21, 3.22 and 3.23 show the evolution of components of the normals for the left arm plane, right arm plane head plane. To compare visually, the trajectories of these are close than its counter part, the joint trajectories.

In order to evaluate the quality of the transfer other kind of metrics must be adopted, this is a study that we have not explored widely. For example, this could be interesting to have a metric that allow us to measure the quality of the same motion over several subjects and compare it with the performance of the motion transference. In [Hersch et al. 2008] the problem of defining other kind of metrics has been studied. In this study instead of evaluate the quality of the motion reproduction of a single trial (joint trajectory based), several examples are recorded and the metric is defined in respect of the rate of success of the task in different conditions. However, this method requires learning and is specific for a single task.

3.4 Conclusion

In this Chapter, first, we discussed human motion representations. Secondly, the Human Normalized model was proposed and developed. This representation has been validated to show that is a viable model to serve as vehicle to transfer motions to kinematic chains. The model includes head, waist, and chest information which in some extent codifies the coordination between these body parts that is present in human when walking or reaching tasks. These motion properties are important information we chose to be transferred to virtual characters or to humanoids. Finally, it was revealed the importance on the representation to evaluate and compare similarity between motion.

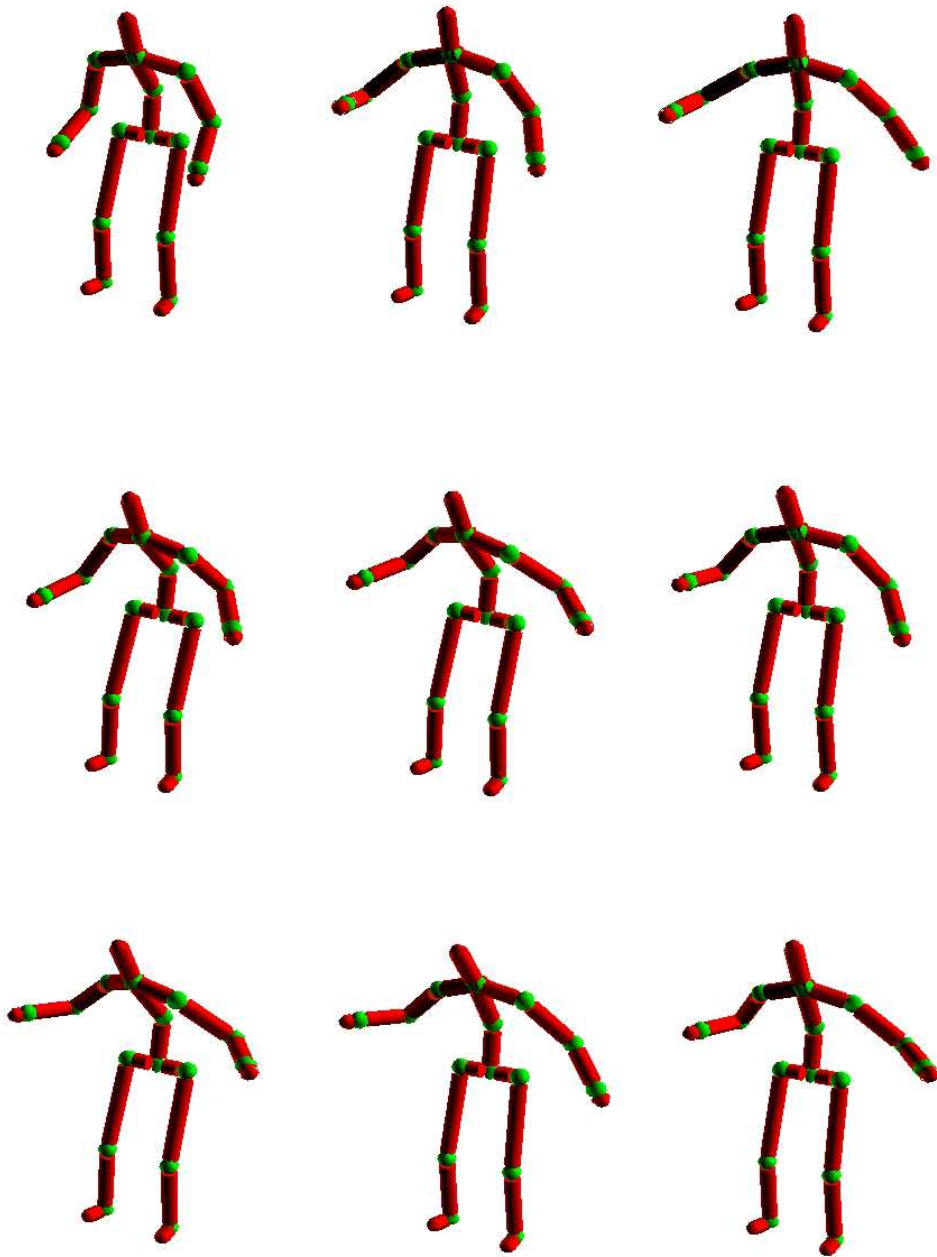


Figure 3.17: Snapshots of the reference character motion, this animation is the result given by the motion capture system.

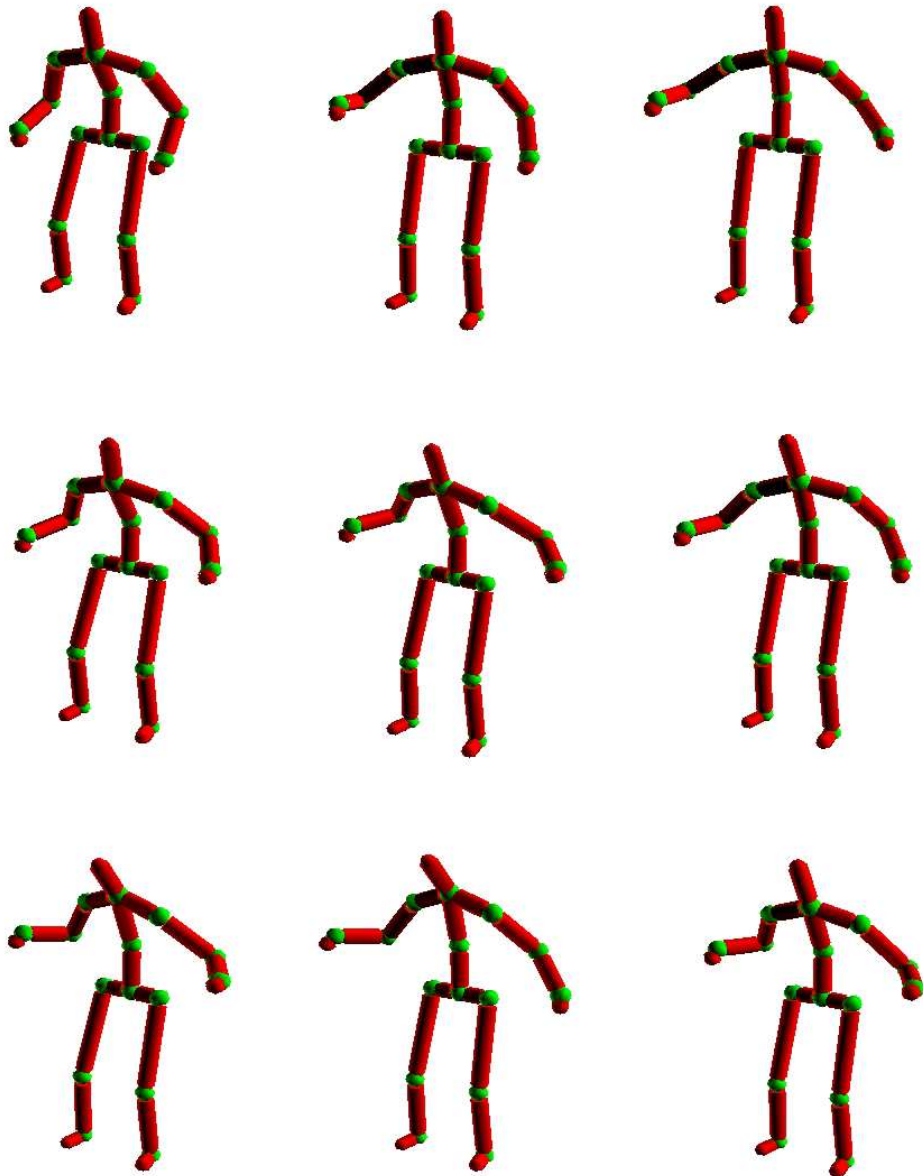


Figure 3.18: Snapshots of the skeleton driven for our HN model and inverse kinematics, same frames as the reference sequence Figure 3.17.

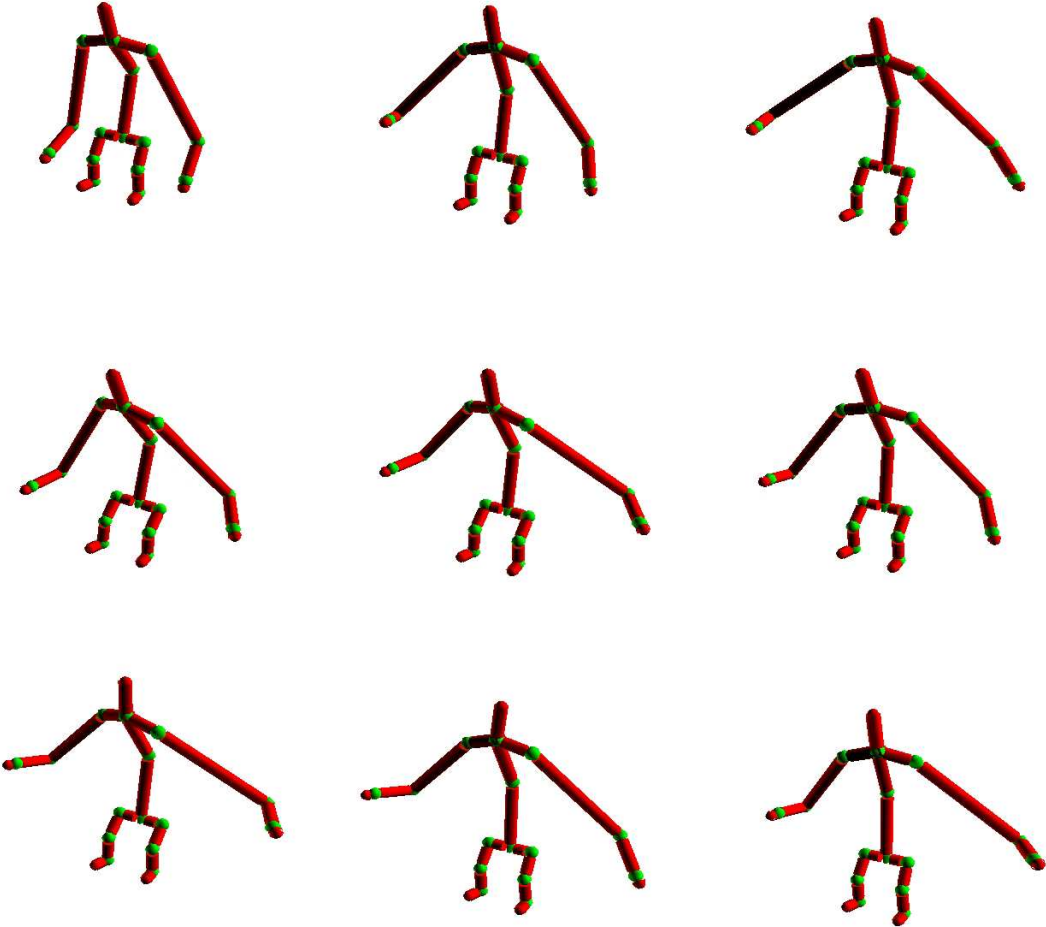


Figure 3.19: Sequence of a character with different arm and legs sizes, same frames as the reference sequence Figure 3.17.

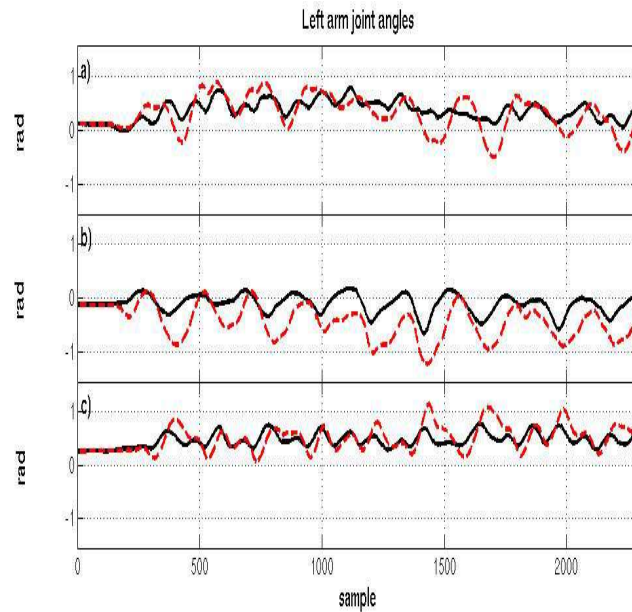


Figure 3.20: Comparison between joints: solid black line are the joint of the reference avatar, dashed lines are the joint values generated by our method.

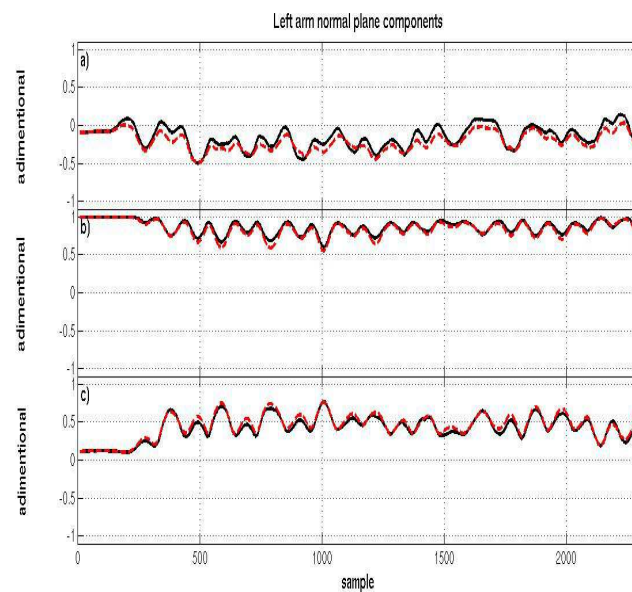


Figure 3.21: Comparison between left arm normal plane: solid black line represents the reference normal components, dashed lines normal components generated by our method. These trajectories are close each other than the reference angles.

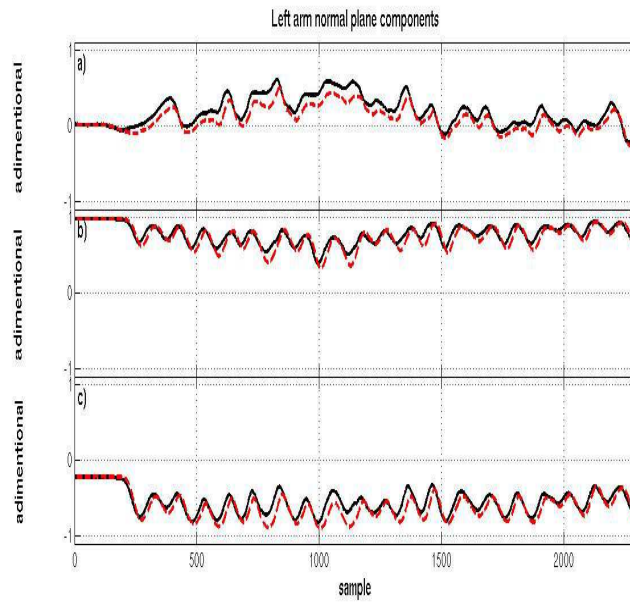


Figure 3.22: Comparison right arm normal plane: solid black line represents the reference normal components, dashed lines normal components generated by our method.

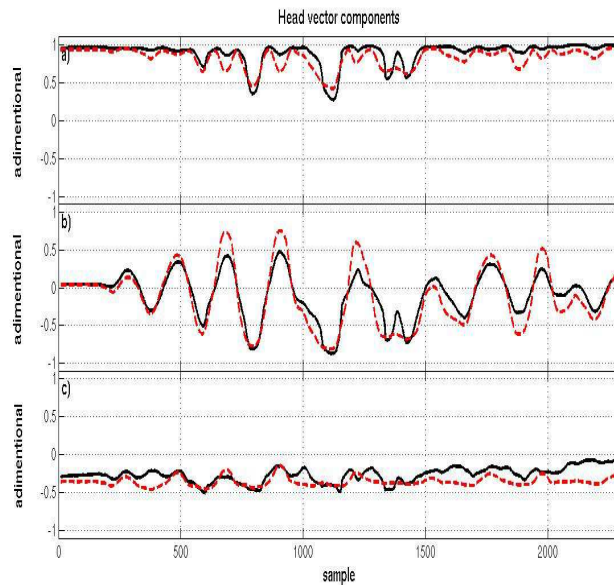


Figure 3.23: Comparison head orientation vector.

4

Online Human-Humanoid Imitation

In this Chapter we present an online framework to transfer human motion to HRP2 [Kaneko et al. 2004]. It is based on prioritized inverse kinematics and the HN model representation. As mentioned in Chapter 4, HRP2 is an humanoid with 30 degrees of freedom, 58 kg and 1.54 m tall and has been manufactured by Kawada Industries, Japan. This robot has human aspect that could be analysed as a kinematic tree with multiple end-effectors, i.e. hands, feet and head. As an example in Figure 4.1 we represent the robot with performing several tasks at the time. The robot is being controlled by a head task, a task for each hand, and a left foot task, additionally the balance of the robot is taking into account by a center of mass task.

4.1 Upper Body Imitation by HRP2-14

Figure 4.2 is depicted the overall method. It comprises four stages:

1. The seamless human motion capture: the objective is to capture the marker position of the reflective markers attached to the body of the human performer and to fill the gaps of the trajectories of the markers when the system cannot determined it. Also, the marker trajectories are smoothed by a low pass filter.
2. Humanoid Normalized model: from the marker position the entries of the HN represen-

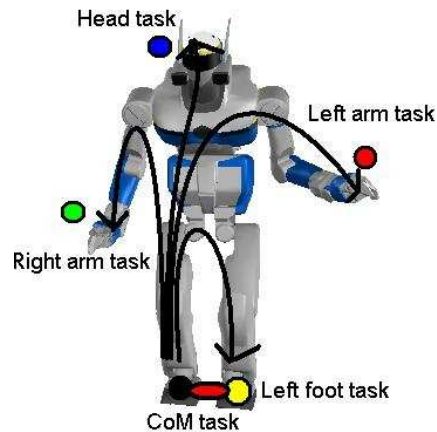


Figure 4.1: Robot HRP2 performing five tasks, the black lines shows four kinematic chains and the red ellipse represents CoM task.

tation are computed. The center of mass trajectory is computed by the proposed com anticipation model.

3. Whole body motion generation: a stack of task to control the robot has been predefined. To generate the joint trajectories of the robot this stack of tasks is solved by prioritized inverse kinematics. The target entries of the stack of task are the parameters of the HN model.
4. Execution on Humanoid: the computed joint trajectories and reference ZMP are feed to the controller of the robot. The stabilizer of the robot is turned on to balance the robot and avoid it falls. In this stage the balance criteria (ZMP location) and the joint limits of the robot are supervised to guarantee that are satisfied. In the case, they are not fulfilled we stop transfer loop.

The Zero Momentum Point (ZMP) has been proposed by Vukobratovic [Vukobratovic and Stepanenko 1972] for a flat ground. In a biped robot additionally to the actuated joints that are controllable directly, there exists a passive degree of freedom between the foot contact and the ground. The foot contact is very important to the walk realization because the robot's position with respect to the environment depends on the relative position of the foot (or feet) with respect to the ground. The foot (contact) cannot be controlled directly but in an indirect way, by generating proper dynamics of the robot above the foot. The ZMP is the point where the influence of all forces acting on the robot can be replaced by a single force. The support polygon is the convex polygon including all the points of the feet in contact with the ground. If the ZMP is inside the supporting convex polygon the system is in dynamic equilibrium. So the basic task to control the robot without fall is to keep the ZMP within the support polygon. Depending on the model used to represent the robot different relations exists [Kajita et al. 2009].

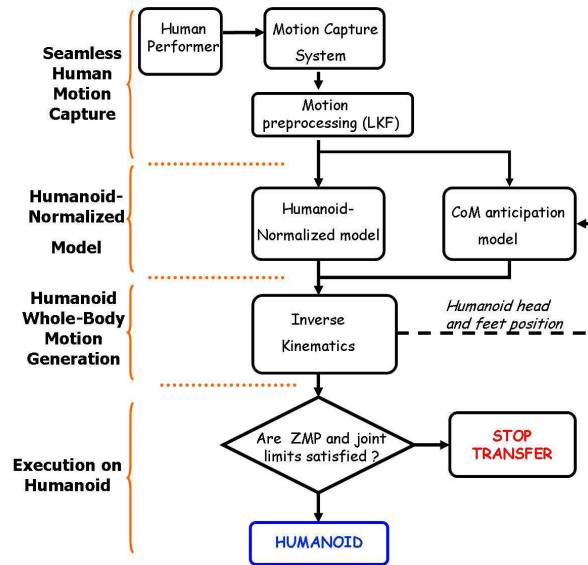


Figure 4.2: Organization of the algorithm to enable real-time motion transfer from the human performer to the humanoid robot.

For example, if the robot is represented a point with mass m the ZMP (p_x, p_y) is defined as

$$p_x = x - \frac{(z - p_z)\ddot{x}}{\ddot{z} + g} \quad (4.1)$$

$$p_y = y - \frac{(z - p_z)\ddot{y}}{\ddot{z} + g} \quad (4.2)$$

$$(4.3)$$

where (x, y, z) and $(\ddot{x}, \ddot{y}, \ddot{z})$ are the position and acceleration of the position of the point mass representing the robot, $p_z = 0$ for a flat ground, g is the gravitational acceleration.

4.2 Seamless Human Motion Capture

The marker positions tracked by the motion capture system are easily recovered for online processing and analysis. However, these raw data cannot be directly used because they are noisy and sometimes some of them are not identified by the system. To fill these gaps we implemented a Linear Kalman Filtering to predict the position of a marker in the instants that the position of that marker is not identified. We employ the following linear system of motion for this purpose,

$$s_{t+1} = A s_t + v_t \quad (4.4)$$

this equation represents the discret system to be analyzed, s_t is the state of the system, A is the transition matrix and v_t is a noise term. The noise in the state of the system is assumed to be Gaussian with zero mean and covariance Q ,

$$N(0, Q) \approx p(v)$$

The equation representing the measurements of the system states is given by

$$m_t = H s_t + w_t \quad (4.5)$$

where m_t are the measurements, H is a matrix relating the state of the system with the measurements, w_t is the noise in the measurements. This noise in the measurements is assumed to be Gaussian with zero mean and covariance R .

$$N(0, R) \approx p(w)$$

For this linear system, the filter of Kalman determines the conditions that minimizes the covariance between the estimated states \hat{s} and the real states s , that is $P = (s - \hat{s})(s - \hat{s})^T$ is to be minimized. The solution of this problem is the recursive Kalman filtering. For each iteration two steps are needed. First, the time update step determines the expected value of s_{t+1} , and the associated covariance matrix P_{t+1} . The prediction for m_{t+1} can be then calculated from the expected value of x_{t+1} . Second, the measurement step update use the measured m_{t+1} to determine x_{t+1} and P_{t+1} for the next recursive step. The time update step is given by

$$\hat{s}_{t+1|t} = A \hat{s}_{t|t} \quad (4.6)$$

where $\hat{s}_{t+1|t}$ is the expected value of s_{t+1} given $m_0 \dots m_t$. Its associated covariance is computed from

$$P_{t+1|t} = E[(s_{t+1|t} - \hat{s}_{t+1|t})(s_{t+1|t} - \hat{s}_{t+1|t})^T] = A P_{t|t} A^T + Q \quad (4.7)$$

The expected value of the measurements (prediction) is given by

$$E[m_{t+1|t}] = E[H x_{t+1|t} + w_{t+1|t}] = H \hat{s}_{t+1|t} \quad (4.8)$$

From the measurement m_{t+1} and its expected value, the measurement step update is achieved by

$$\hat{s}_{t+1|t+1} = \hat{s}_{t+1|t} + K_{t+1}(m_{t+1} - H \hat{s}_{t+1|t}) \quad (4.9)$$

$$P_{t+1|t+1} = P_{t+1|t} - K_{t+1} H P_{t+1|t} \quad (4.10)$$

where K_{t+1} is the Kalman gain given by

$$K_{t+1} = P_{t+1|t} H^T (H P_{t+1|t} H^T + R)^{-1} \quad (4.11)$$

To predict the marker positions we assume that the markers are moving with constant acceleration. The system with constant acceleration is given by

$$\begin{aligned} p_{k+1} &= p_t + v_k \Delta T + a_k \Delta T^2 / 2 \\ v_{k+1} &= v_t + a_k \Delta T \\ a_{k+1} &= a_k \end{aligned} \quad (4.12)$$

where p , v , a are the position, velocity and acceleration of the maker; and ΔT is the time step. In practice the constant acceleration was determined from the media acceleration of all makers in a typical human motion session. The covariance of the noise are taken from the motion capture calibration step.

4.3 Whole Body Motion Generation

The position for each joint is generated from a prioritized stack of tasks [Yoshida et al. 2006], which is solved using the damped prioritized inverse kinematics formulation. Each entry in the HN model representation is used as the target input for a task. That is, the problem of imitating or transferring human motion to a robot is the problem of solving a predefined set of tasks having the HN elements as targets.

Our task stack we use to transfer motion is defined as (in decreasing priority):

1. Homogenous transformation task for each feet, i.e. both position and orientation are fixed,
2. Position task for Center of Mass (CoM) projection (X and Y positions),
3. Position task for the head,
4. Homogenous transformation task for the left wrist,
5. Homogenous transformation task for the right wrist,
6. Orientation vector task for the chest,
7. Orientation vector task for the waist,
8. Orientation vector task for the head.

We use four kinds of tasks: position task, orientation vector task, homogenous transformation task and a CoM task. As examples, we define in more detail the task construction for the

head and the arms. The CoM task is detailed in the next subsection. The position task of the head $f_h(\theta)$ is defined as,

$$f_h(\theta) = P_h^t - P_h(\theta)$$

where P_h^t is the target of the task given by the position of the head in the HN model representation. $P_h(\theta)$ is the position of the humanoid head expressed as a function of the robot dof's θ .

For the orientation vector task of the head $f_h(\theta)$ we have

$$f_h(\theta) = V_h^t \times V_h(\theta) \quad (4.13)$$

where V_h^t is the target head direction, which is given by the head orientation vector in the HN model. And $V_h(\theta)$ is the corresponding vector of the humanoid head, as a function of the robot dof's θ . The orientation vector tasks for the chest and waist are defined in a similar way. For each arm a homogeneous transformation task is constructed for the wrist joint. That is because, we want to preserve the three dof's in the rotation. The target transformation is constructed from two properties, the wrist position and the normal to the HN model's arm plane. The target rotation matrix in the homogeneous transformation is computed as,

$$R^t = [N_{arm} \times V \quad N_{arm} \quad V] \quad (4.14)$$

where N_{arm} is the normal of the left or right arm plane, and V is a unit vector connecting the elbow and wrist markers. For the HRP2 robot, it should be noted that the N_{arm} is parallel to the axes of the humanoid elbow joint and the wrist joint. To represent the orientation we use the matrix representation over others representations mainly because of architecture compatibilities.

4.4 Center of Mass Anticipation Model

The Center of Mass of a humanoid robot is a strong indicator to its stability. In order to remain statically stable, the projection of CoM on the floor should remain within the support polygon defined by the two feet of the humanoid. If the human performer were to lift his/her foot, the CoM of the humanoid robot would have to be shifted in advance towards the other foot in order to maintain balance. In order to know when this shift is required, we take inspiration from results in human neuroscience research. Studies have reported strategies by which motion of the CoM in humans can be related to foot placement and hip orientation [Patla et al. 1999], [Sveistrup et al. 2008].

To manipulate the projection of the humanoid CoM on the floor we constrain it to track a target. The target position is computed depending on the current stance of the HN model, i.e.

Double Support (DS) or Single Support (SS). The transition of stance from single to double support is detected using the position and velocity of the feet. When either of these measures exceed a pre-determined threshold a change of stance is said to have occurred.

For the CoM task motion, its target is computed as:

$$CoM_i = \begin{cases} CoM_{i-1} + \alpha(V_{head} \cdot V_{feet})V_{feet} & \text{if DS} \\ p_{foot} + \beta V_{head} & \text{if SS} \end{cases} \quad (4.15)$$

where,

CoM_i = CoM X and Y positions at time step i ,

V_{head} = HN Model head 2D velocity vector,

V_{feet} = Unity vector across robot's feet, i.e. the unit vector joining from left foot to right foot of the robot,

p_{foot} = Humanoid support foot X and Y positions,

α, β are constants.

This model is the implementation of the ideas in [Patla et al. 1999] and [Sveistrup et al. 2008] where the center of mass projection on the ground is studied from the footprints, head motion and hips motion. In practice α is related with the scaling factors between the human height and the robot. It represents the influence of the head motion on the CoM projection in DS. β is the influence of head motion on the CoM projection in SS. It has a low value because of the stability: we want the CoM to remain on the support foot sole.

4.5 Tests and Evaluation

We present two scenarios that illustrate the capabilities of our algorithm. In the first scenario we assume the robot's feet to be fixed and imitate the motion of a human performer executing a slow dance with the upper body, including bending of the knees and ankles. In the second scenario, the robot is allowed to lift-off with one of its feet and balance on the other foot. This was chosen to illustrate the anticipation model which prepares the humanoid for balancing on one foot. The parameters used for the CoM anticipation model were, $\alpha = 0.12$, $\beta = 0.01$. All computations were run on an Intel Core 2 CPU 6400 @2.13GHz, with 2GB of RAM memory. At each solution step we required ~ 30 ms to build the Humanoid-Normalized model and to solve the stack of tasks using a damped inverse kinematics solver. At the end of the Chapter some sequences of online transfer human motion are shown in Figures 4.12 4.13 4.14 4.15.

4.5.1 Practical Implementation

Our framework was implemented using the software architecture Genom, [Fleury et al. 1997]. Mainly, we have two modules to establish communication from the motion capture system to

the HRP2 robot interface. First, we have a motion capture server whose function is to send motion data to the network via UDP protocol. First, a Linear Kalman Filter is used to predict markers positions in case where some of them are not identified by our motion capture system. These data are then filtered by a Butterworth low pass filter. A second Genom module reads the seamless motion data and computes robot motion. This module implements a HN-normalized model, a CoM anticipation model and a prioritized inverse kinematics solver. Finally, via a plugin, we send robot motion data from the motion generation Genom-module to the HRP2 interface control panel.

The Humanoid Robot HRP2 includes a realtime stabilizer module. This module is based on reference Zero Moment Point trajectories, foot contact forces, internal central unit measurements. It modifies the the feet joints to balance properly. However, it is provided as a blackbox.

4.5.2 Dancing

The human performer was asked to perform a simple dance without stepping or sliding his feet. Figure 4.3 shows the posture of the human and the humanoid in the middle of a dance. The motion computed by the algorithm was smooth, without joint position or velocity limit violations, and was quasi-statically stable.



Figure 4.3: Scenario 1: snapshot of human dancing and its imitation by HRP2.

Figures 4.4 and 4.5 show roll, pitch and yaw angles of the head and chest of the HN model and those of the humanoid robot. Despite low priority given to the head orientation task, we observe that yaw and pitch angles were matched very closely, while roll angle of the humanoid was much lesser than the HN model. This was because yaw and pitch axis are directly available on HRP2 (independent of the other joints), however, the humanoid does not have a roll axis for the head joint. The roll variation seen in Figures 4.4 was due to the movement of the whole body. That is, the head position and the center of mass tasks have higher priorities which produced motion on the legs to bend the body towards the head and CoM targets. This produces a roll motion on the upper body.

Chest roll of the HN model was not considered, but to account for the movement of the rest of the body we see an induced roll component on HRP2. The pitch and yaw angles of the humanoid’s chest followed the HN model less closely due to the lower priority of this task. Since the arms and the head are connected to the chest, and their respective tasks have a higher priority, the chest joint has a reduced degree of mobility.

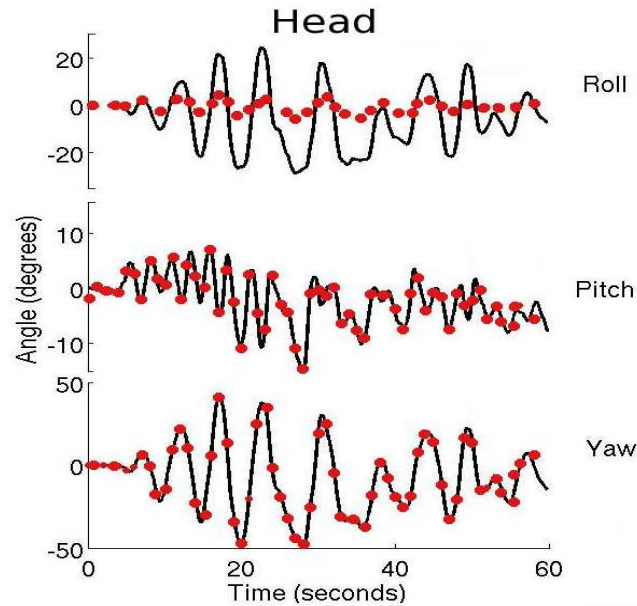


Figure 4.4: Roll, pitch and yaw angles of the head joint during the dancing motion. Solid black line indicates the angles of the HN model, this was the target the humanoid had to follow. Red circles indicate the corresponding angular value sent to HRP2. The roll motion of the head is induced from the legs motion.

4.5.3 Foot Lift

In this scenario the human performer shifted his weight onto one leg and maintained his balance for a few seconds before slowly returning to rest on both feet. Figure 4.6 shows the human and the humanoid balancing on one foot (SS stance).

The head motion in the HN model, and the vertical position of the foot of the humanoid is plotted in Figure 4.7. We observe that the head shifts towards the support foot (right foot) before the other foot lifts (Figure 4.7). The sideways displacement of the head reaches the Y position of the support foot about 1s before foot lift. Before reaching this point, CoM projection was derived according to Equation 4.15 (DS stance). Once the head reaches the support foot, the CoM is maintained at this position (referred to as “Balance Point” in Figure 4.7). After this point, the behavior of the CoM is dictated by a different relation (SS stance in Equation 4.15). It should be noted that for slow head motion, the projection of the CoM and the head position coincide (a small offset can be seen in the zoom inset in Figure 4.7).

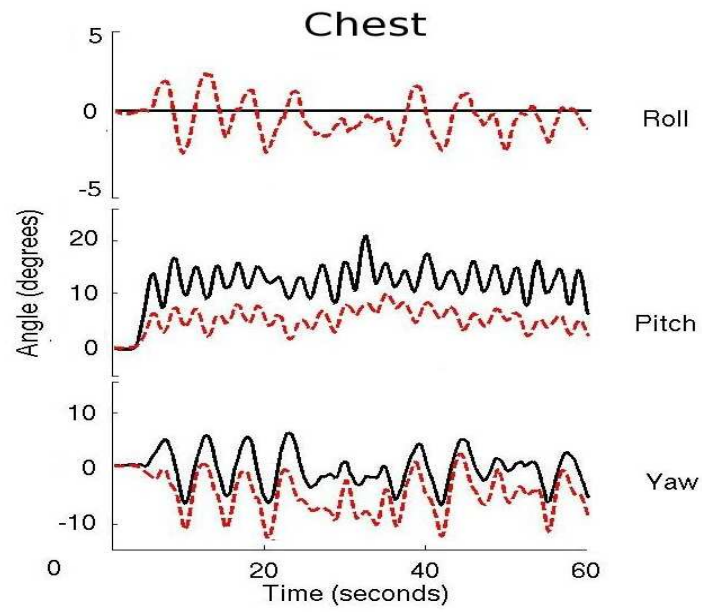


Figure 4.5: Chest angles of the HN model (solid black line) and the corresponding angles on the humanoid (dashed red line).



Figure 4.6: Scenario 2. Picture of humanoid imitating the human lifting his foot.

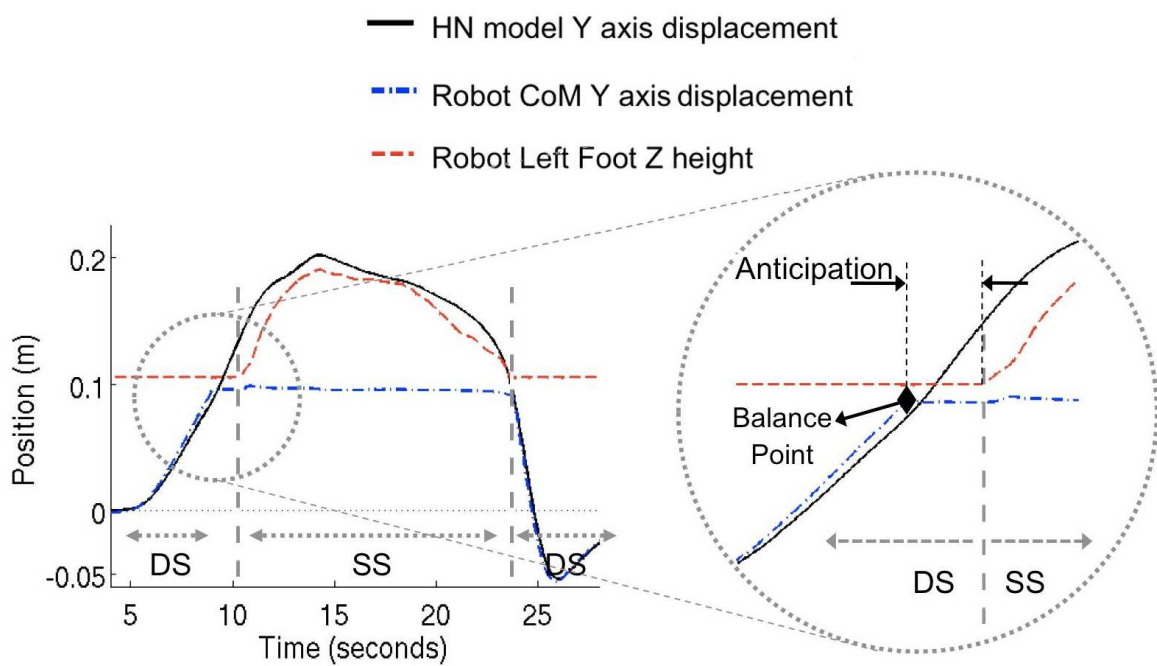


Figure 4.7: Sideways displacement of the HN model head (solid black line), and the vertical height of the humanoid’s lifting foot (dashed red line). Also shown is the Y displacement of the humanoid’s CoM (dash-dot blue line). Zoom inset shows a magnified view of the anticipation phase. The anticipation occurs between the ‘Balance point’ and the time when the humanoid lifts its foot.

4.5.4 Quality of motion imitation

Quality of motion imitation was quantified by measuring the root mean square error between the target (HN model) and the humanoid robot. Table 4.5.4 lists the relevant parameters and its corresponding errors. The position of the CoM and head were tracked almost perfectly. This was because both these tasks had a very high priority. Comparatively, head orientation which had a lower priority had a mean error of 4 deg. But it should be noted that most of this error was because of the roll angle (HRP2 does not have a head roll axis). The right wrist position error was slightly larger than the left wrist. This can be attributed to the fact that left wrist task came before the right wrist task in the priority list. Thus, once the left wrist position and orientation was fixed, it became more difficult for the right wrist to reach exactly its target transformation. Comparing across studies, [Dariush et al. 2008], reported an error of about 0.02 m in tracking the wrist position while assigning them to a “medium priority group”. In our case the head position had the highest priority, and hence a low error, while the hands had low priority, hence larger error. Chest and waist orientation were lower in the priority list and hence show larger errors in orientation than other joints. Overall, these results show that we were able to retarget a large part of the motion of the human onto the humanoid.

Property	Mean RMS position (m)	Mean RMS orientation (deg)
CoM	~ 0	-
Head	~ 0	4.09 (R:11.8 P:0.22 Y:0.26)
Left wrist	0.02 (X:0.013 Y:0.02 Z: 0.3)	-
Right wrist	0.05 (X:0.05 Y:0.017 Z: 0.08)	-
Chest	-	4.2 (R:1.1 P:6.7 Y:4.8)
Waist	-	6.4 (R:1.1 P:4.62 Y:0.52)

Table 4.1: Mean RMS error between HN model and humanoid. Values in brackets denote the mean RMS error in X,Y and Z positions for wrist positions, and roll (R), pitch (P) and yaw (Y) for head, chest and waist orientations

4.5.5 Transfer Motion Limitations

In order to know the limits in our approach we develop two evaluation test. First, we determined what are the maximum speed of the motions that we are capable of transfer motion to the humanoid with our approach. For this purpose, we attached the marker set on the performer

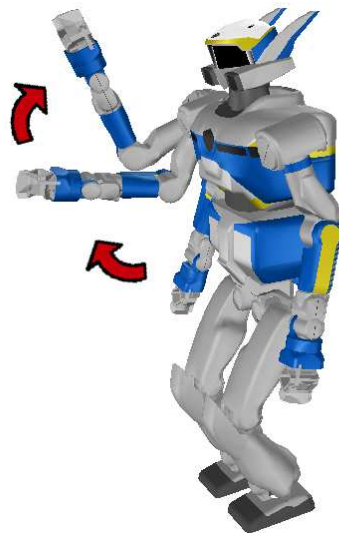


Figure 4.8: Illustration of the hand up and down motion on the simulated HRP2.

and he was asked to move his right hand in an up-and-down motion at different speeds Figure 4.8. After transferring the motion to the humanoid, we observed the shift of the Zero Moment Point (ZMP) of the humanoid for different human hand speeds. For up and down motions, we detected the maximum and minimum values of the ZMP components, and check if the ZMP is inside the supporting polygon. We found that the humanoid became unstable when hand speed was higher than 1 m/s (Fig. 4.9, 4.10). This example illustrates the limitations of using our inverse kinematics approach.

Because our approach does not includes mass and inertial properties of the robot, automatically it constrains to transfer only moderate speed motions. The inertial properties become important when fast motion are to be transferred. To imitate more accurately both aspects the kinematics and dynamics of the human motion would be taken into account during the modeling stage. For example, using a dynamical model (exact or simplified) at motion planning stage (HN model computation) could be a useful in this regard.

Secondly, we asked how fast motion by our approach could be performed if a dynamic model of our robot is used, at expense of the computation time. We recorded a dynamic motion, the petanca motion, and tried to reproduce it by our approach. We found that the 7.2 seconds, that takes place the petanca shot was impossible to transfer directly to the robot. In order to be reproduced by our method, we increased the length time of the motion to 22.5 to obtain a viable solution, In Figure 4.11, we show a slow version of the petanca shot motion being played by HRP2 in simulation.

In order to determine how fast this trajectory can be played on we proceeded as follows: this trajectory was used as the input for an optimization based approach that accounts for the dynamic model of the robot [Kanehiro et al. 2008]. This method uses a simplified dynamic model, the model car-table [Kajita et al. 2003], and includes constraints on the ZMP location.

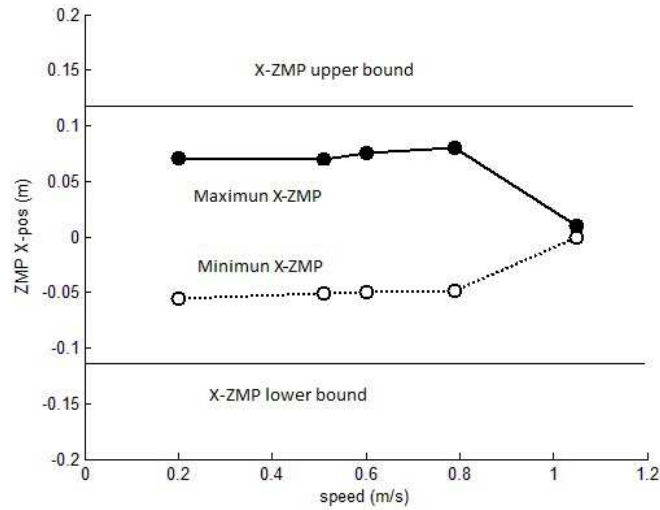


Figure 4.9: Plot of X component of ZMP of the humanoid robot vs. human hand velocities. Illustrated are the bounds.

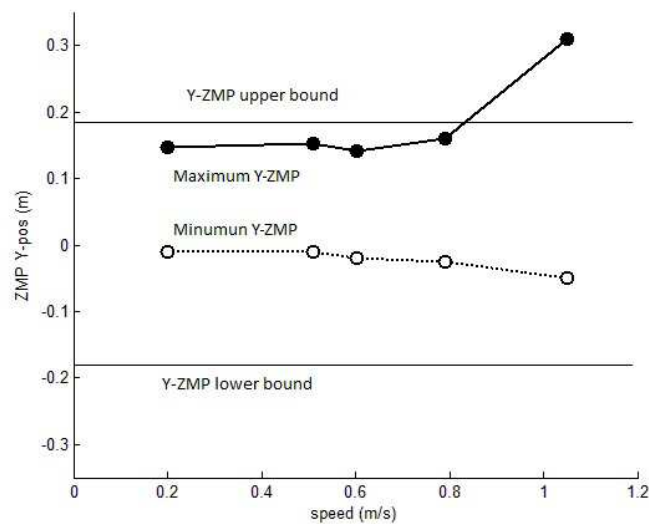


Figure 4.10: Plot of Y component of ZMP of the humanoid robot vs. human hand velocities. Illustrated are the bounds and a point where the robot becomes unstable.

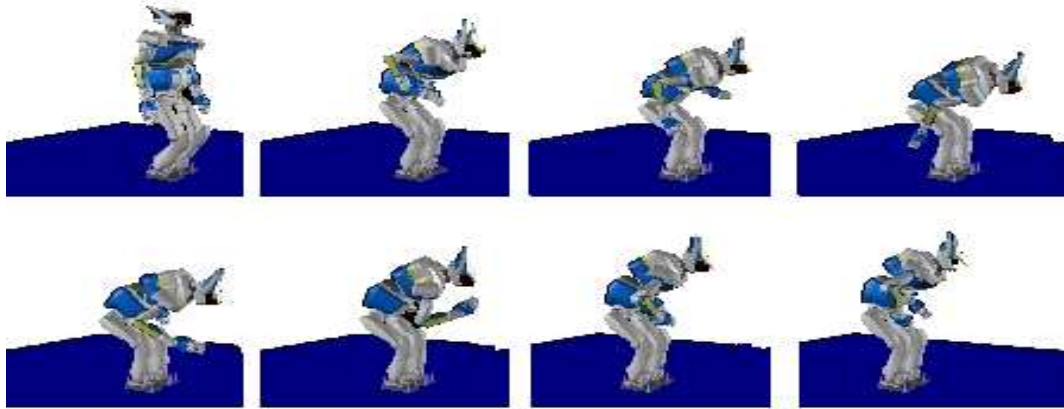


Figure 4.11: Sequence of the petanca motion played on the OpenHRP3 simulator by HRP2.

It determines the shortest time trajectory that is viable for the humanoid. We found that this motion was possible to be played in 6.8 seconds. However, the method takes 12 minutes to compute the final motion.

4.6 Conclusion

In this Chapter we presented an online framework to transfer human motion towards the humanoid robot HRP2. First, the HN model was used to serve as vehicle to transfer human motion to the humanoid. Its parameters were used as input for an inverse kinematic solver to generate robot motion. Even if our approach is based on kinematics it allows us to transfer an ample range of motions. That is mainly due to the Center of Mass task we included to generate the robot motion and to the autobalancer of HRP2. Second, this approach has its limits, that was the reason we also presented some tests aiming to evaluate its performance and limits. Finally, not less important from the practical view of point, we discussed some issues related to the online motion transfer. For all our experiments, self-collisions tests were not performed, but almost in practice we did not find big troubles in this critical issue. Even if not presented, we performed some tests on auto-collision, but these should be managed as inequality constraints. This issue increased the latent time of our solver which is not practical for online purposes. The self-collision was an issue that was supervised by ourselves, i.e. if we observed a risk of collision we stop the transfer on the robot.

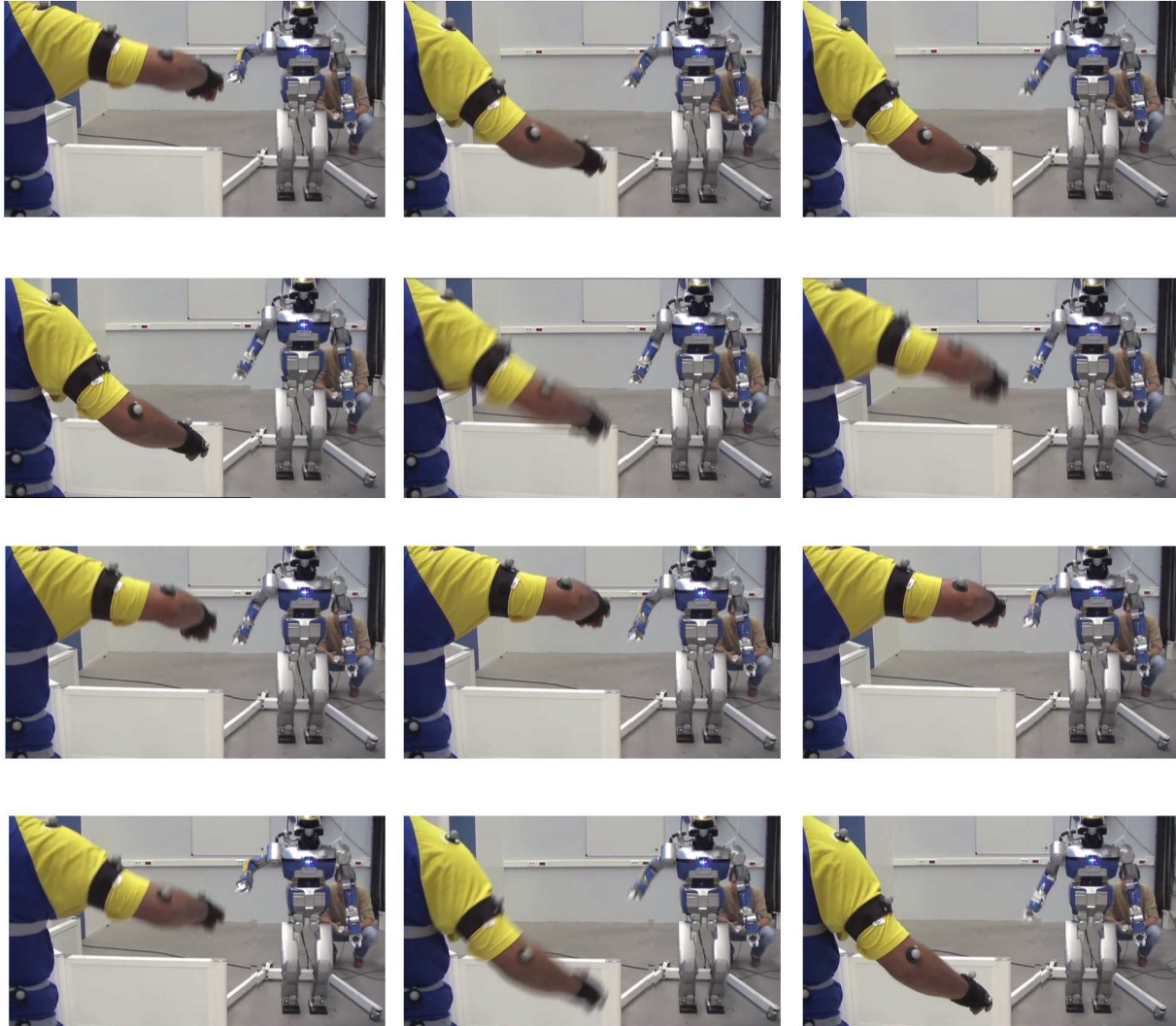


Figure 4.12: Arm sequence: this sequence show us that the robot imitates the arm motion that was modeling as a normal vector and a position. In practice it is very important to place properly the reflective markers on the body, we must avoid to place the markers in such a way they could become co-linear (no plane is uniquely defined).

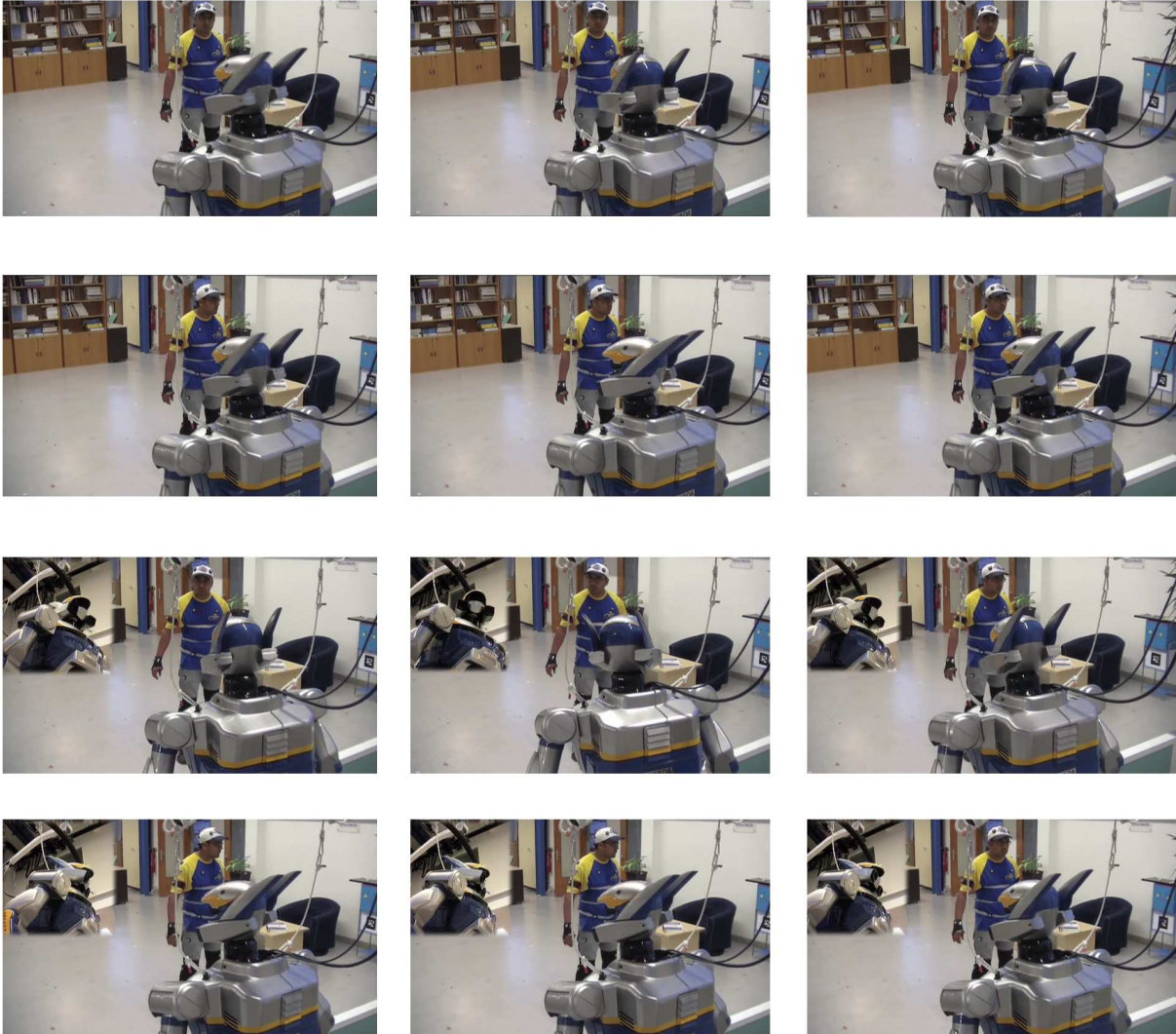


Figure 4.13: Head and chest sequence: this sequence show us the transfer of the chest and head human movement to HRP2. Another source of problems in practice is the lighting in the motion capture room. The motion captures system cannot adapt to these changes resulting in bad performance in tracking and identifying the marker set.



Figure 4.14: Dancing motion: in this sequence the motion include movement on the head, chest, arms and waist. Here, the kalman filter is very important because arms hidde some markers to the system. Its position is estimated by the filter.

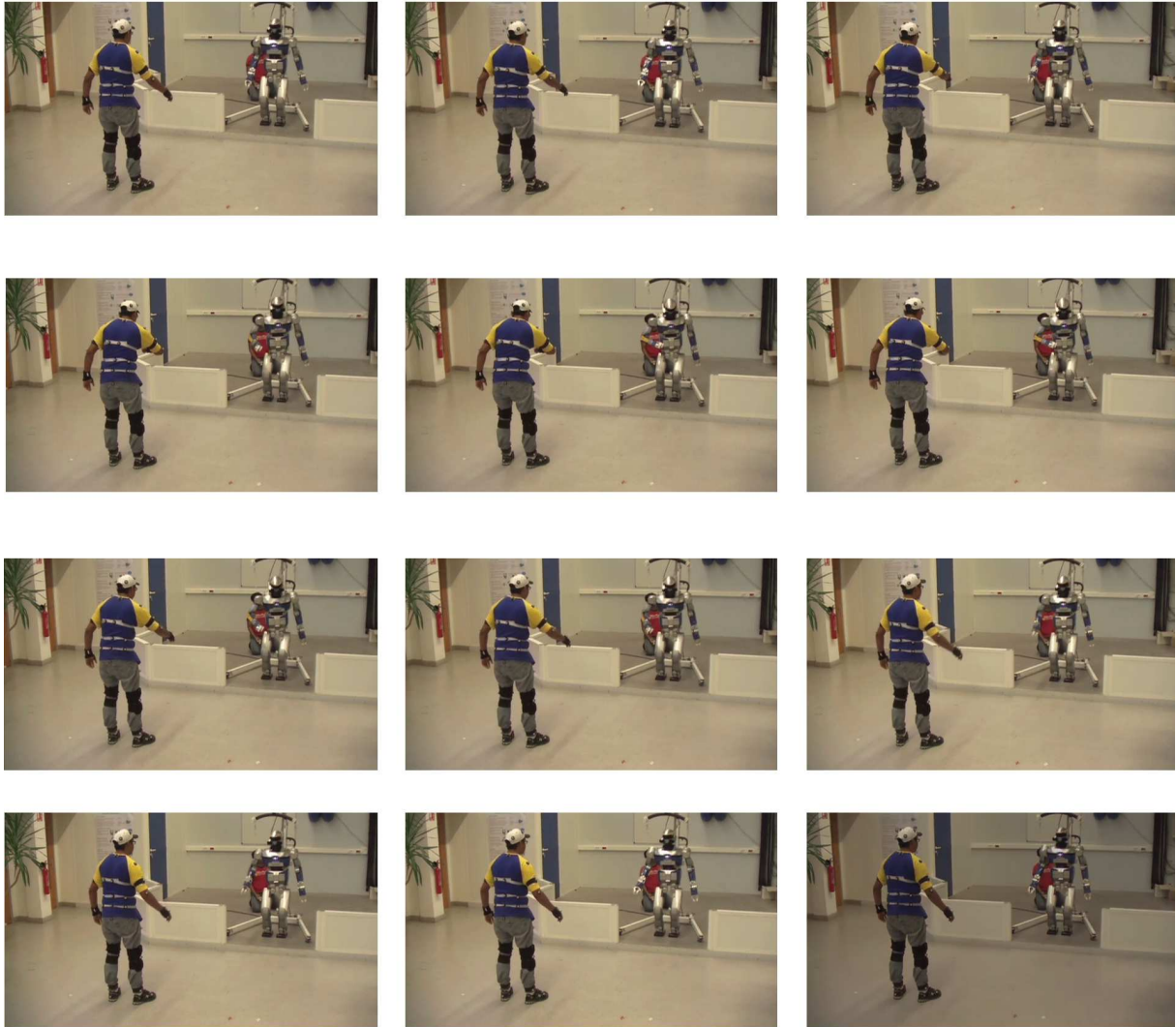


Figure 4.15: Bow: this sequence shows HRP2 performing a bow.

5

Imitating Human Motion: Including Stepping

In this chapter we study the problem of transferring both feet motion and upper body motion by the humanoid robot HRP2. This problem is more complex than transferring upper body motion with feet fixed, because of the balance problem when feet displaces.

Human beings displace in the environment by moving its feet in particular patterns. Certainly, the upper body motion also contributes to this displacement, i.e. studies has been reported about anticipation of head and trunk when humans walks [Sveistrup et al. 2008]. These particular patterns are used to classify the human motion in walking, running, double support stance and single support stance. However, there exists some behaviors in which humans use their feet for other purposes, for example, dancing or reaching. These motions are difficult to classify because depends on the intention of the overall motion and the environment context.

Among all these such large range of motions using feet, walking as been well studied and successfully implemented on humanoid robots [Lim and Takanishi 2007] [Hirose and Ogawa 2007] and recently [Kim et al. 2009]. It is classified into passive, static and dynamic. Passive walking is achieved by machines built from passive elements, the power required for walking comes essentially from gravity, some studies on this subject has been reported in [McGeer 1990] [Garcia et al. 1998].

In the following paragraphs we develop an overview on static and dynamic walking. First, we introduce some terms:

- an humanoid is in double support (DS) when both feet are in contact with the ground,
- an humanoid is in single support (SS) when one feet is in contact with the ground,

- the foot which is in contact with the ground in SS is called supporting foot, the other foot is called swinging foot,
- the convex area defined by all the points of the feet in contact with the ground is called the supporting area.

Over a flat ground, static walking refers to a system which stays balanced by always keeping the center of mass (COM) of the system vertically projected over the polygon of support formed by the feet. Whenever a foot or leg is moved, the COM must not leave the area of support formed by the feet still in contact with the ground. Static walking is realized in the following way:

1. assume an humanoid is initially in DS, the robot shifts its the center of mass projection to one foot (the supporting foot),
2. the robot keeps its center of mass projection on the sole of the supporting foot, while the other foot (swinging foot) is displaced to a footprint target,
3. the robot keeps its feet fixed to the ground and displace its center of mass projection from the actual supporting foot to the sole of the actual swinging foot,
4. the roles of supporting foot and swinging foot are interchanged, repeat points 2) to 4).

In order to maintain static equilibrium throughout the walking, speed limits are imposed on the humanoid robot motion to minimize the inertial forces. One characteristic of this walk is that robot does not fall if the motions is stopped suddenly at any time. However, even if the planning of one static step is very fast, the physical performance requires a considerable amount of time.

On the other hand, during dynamics walk faster steps are performed. This walk uses the actuators of the robot and ground contact effects of to ensure propulsion. During dynamic walking the COM may leave the support area formed by the feet for periods of time. This allows the system to experience tipping moments, which give rise to horizontal acceleration. However, such periods of time are kept brief and strictly controlled so that the system does not become unstable. Thus one may think of a dynamically balanced system as one where small amounts of controlled instability are introduced in such a way as to maintain the overall equilibrium

The dynamic walk is stable if it is sustainable without fall and it allows safe return to a statically stable configuration. In order to evaluate the dynamic stability of walking robots some indices has been proposed: zero momentum point [Vukobratovic and Stepanenko 1972] [Kajita et al. 2003], linear inverse pendulum [Kajita et al. 1990] [Sugihara et al. 2002], free rotation indicator point [Goswami 1999], zero rate of change of angular momentum point [Goswami and Kallem 2004], among others [Foret et al. 2003] [David et al. 2005]. However, the ZMP index is the most popular and extensively used in robotics. Even, the robot HRP2 requires ZMP reference trajectories as input.

In general, a pattern generator is a device used to compute reference joint trajectories for the robot which once played on the robot generates walking. The main idea is to plan the position and orientation trajectories, in the Euclidean space, for the feet and the center of mass trajectory that ensures dynamic stability. Next, the joint trajectories are computed in such a way that the planned feet and center of mass trajectories are fulfilled [Kajita et al. 2003].

In the rest of this chapter, first, we develop an approach to transfer human feet motion to a robot offline. In this case, the main advantages are the unlimited expense of time to compute the robot motion and the full human motion information. In order to transfer feet motion to the robot, we analyzed pre-captured data to plan feet position and orientation trajectories, defined a stack of tasks and solved them by using an inverse kinematic solver. Second, we develop an approach used to transfer full body motion including stepping online. The limited expense of time to compute the robot motion and the limited human motion information are the main difficulties to overcome.

5.1 Offline Human Feet Motion Imitation

The main assumptions for offline human motion imitation are: 1) full human motion trajectory is available, 2) the time to computing reference robot motion trajectories is unlimited. Four markers attached to each foot of the subject test are used to imitate feet motion.

In all motion capture recording sessions, 41 markers were attached to its body of a subject test and we gave him instruction to start moving from a standard position, see Figure 5.1. Then, we gave him instructions to perform movement including his feet.

The first step to transfer captured human motion to the robot is to scale the markers positions. They are scaled by using the initial frames. In particular, we scale the markers position by the ratio of the HRP2 height and the performer height.

5.1.1 Feet Motion Segmentation

Feet motion is composed of several support phases (Figure 5.2), for instance a double support phase, a left single support phase, a right single support phase. These phases are determined according to the feet in contact with the ground. The double support phase means both feet are in contact with the ground and the single support phase means one foot in contact with the ground.

The support phase defines the posture stance. If the human is in double support phase, we say the human is in double support stance (DSS). In the same way, the human is in single support phase (SSS) if the human is in single support phase. It is possible that no support phase is detected, in such a case the human is flying. This situation is not considered in our study because the robot HRP2 can not jump or fly.



Figure 5.1: A subject in a motion capture session, he is in the initial standard posture.



Figure 5.2: Snapshots of supporting phases when displacing feet, from left to right: we have a double support stance, single support stance and double support again.

Feet motion is segmented into basic segments by detecting changes in the human stance. A basic segment of motion is defined as the portion of motion between two consecutive changes of stance. We label each basic segment depending on its stances changes. The basic segments of motion we used are

- START_LFSS is the segment from the initial frame to the frame just before left foot stance has been detected.
- START_RFSS is the segment from the initial frame to the frame just before right foot stance has been detected.
- LFSS_TO_DSS is the segment from the frame at the instant left foot stance has been detected to the frame at the instant double support stance has been detected.
- LFSS_TO_RFSS is the segment from the frame at the instant left foot stance has been detected to the frame at the instant right foot stance has been detected. This situation is not common during “normal” walking, but can be present in other situations.
- RFSS_TO_DSS is the segment from the frame at the instant right foot stance has been detected to the frame at the instant double support stance has been detected.
- RFSS_TO_LFSS is the segment from the frame at the instant right foot stance has been detected to the frame at the instant left foot stance has been detected. This situation is not common during “normal” walking, but can be present in other situations.
- DSS_LFSS is the segment from the frame at the instant double support stance has been detected to the frame at the instant left foot stance has been detected.
- DSS_RFSS is the segment from the frame at the instant double support stance has been detected to the frame at the instant right foot stance has been detected.

These basic segments can be used to determine whether the human is performing an step, i.e by detection DSS_LFSS and inspecting if its length time is short enough.

Figure 5.3 shows a plot of height trajectories of feet and its segmentation. The marker used for segmentation is the heel.

At each time either foot loses or recovers contact with the ground, a change of stance is taking place. These changes of stance are used to segment the motions. In this plot, vertical solid lines are used to indicate the instants where left foot loses or recovers contact, a dashed line is used for the right foot. A segment S_i is defined by the frames included into two consecutive vertical lines. These basic segments are labeled according to its changes of stances as mentioned above. From the plot we can observe undesired situations, the segments S_2 and S_4 are the cases when the human is flying.

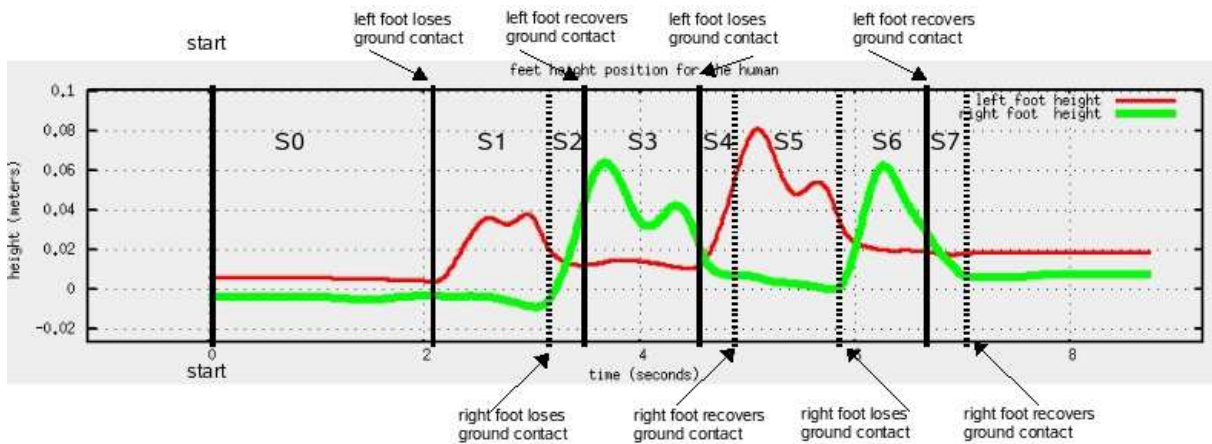


Figure 5.3: Plot of feet height indicating the instants where stance change is detected, S_i are the basic segments detected.

5.1.2 Planning Feet Motion

The feet motion is not imitated but planned. The feet motion is planned by using the footprint sequence detected, Figure 5.4. Furthermore, the ZMP trajectory and the CoM projection to ground trajectory are to be planned to guarantee that robot does not fall. That is, only the footprints of the human motion are planned. In fact an scaled version of the footprints.

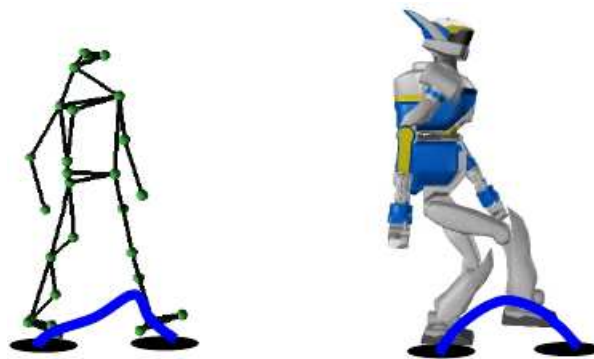


Figure 5.4: Feet motion is planned, it is not imitated directly.

Feet positions and orientation trajectories are to be planned for each basic segment motion, in particular for segments in single support stance. There are two situations to consider. The case in which the segment correspond to walking, i.e. the time length of the segment is around one second, and the case segment time length is larger enough, i.e. the human is in one leg posture for a while.

For the first case, the position trajectory is planned in two parts. First, we plan the horizontal

displacement components of the foot coordinates (x, y) and then the height displacement z .

The horizontal trajectory of the swinging foot are computed from the corresponding foot markers. The positions of the markers are projected to ground. The horizontal trajectory is the average of these projected positions. For the height trajectory, we compute two minimum jerk trajectories, one from 0 to a predefined height and other from the predefined height to 0. We predefined the height of the foot because the HRP2 robot cannot attain positions and to reduce impact on the robot foot at landing instant. The predefined height could be modified as a function of the moving foot height, taking into account the maximal height the robot can attain.

To plan foot orientation only the yaw rotation is considered. We plan it by computing a minimum jerk trajectory between the orientations of the flying foot at the initial and final frames in the basic segment.

In the case the moving foot remains flying for a long period of time (order of seconds), the horizontal trajectory is planned in the same way as explained above. But, the height foot trajectory is planned differently. It is split into two pieces. The first piece is computed as the scaled trajectory of the average z component of the foot marker positions. This piece is from the initial frame of the segment to the frame at the instant there are at least K samples to land. The second piece is a minimum jerk trajectory from the final position of the first piece (with non zero velocity and acceleration) to a 0 position with final zero velocity. The yaw trajectory is planned using the same principle.

5.1.3 Planning Zero Moment Point and Center of Mass

To plan the center of mass trajectory that satisfies the ZMP index of stability we use the preview controller developed by Kajita [Kajita et al. 2003] and the feet information at the initial and final frames in the basic segments. We recall first the preview controller and then develop the method used to plan the center of mass trajectory.

5.1.3.1 Preview Controller

Using a simplified model of the robot, in particular the car-table model, we can determine the relation between CoM and ZMP positions, refer to Figure 5.5. Given the car motion, viewed as the CoM motion, we can determine the ZMP trajectory by using the next equation

$$p_x = x - \frac{z_c}{g} \ddot{x} \quad (5.1)$$

$$p_y = y - \frac{z_c}{g} \ddot{y} \quad (5.2)$$

where (p_x, p_y) is the position of ZMP on the ground; (x, y, z_c) is the position of the CoM.

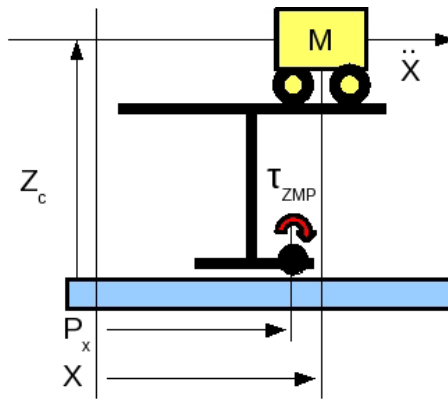


Figure 5.5: Cart-table model, represents the relationship between the ZMP, p , and the CoM, x , positions. τ is the torque at p and M is the robot mass.

The inverse problem of the cart-table model is used to determine the CoM trajectory from a given ZMP trajectory. From Equations 5.1, 5.2 and a control variable defined as $\frac{d}{dt}\ddot{x} = u_x$. The dynamical system of the cart model table is given by

$$\frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_x \quad (5.3)$$

$$p_x = \begin{bmatrix} 1 & 0 & -z_c/g \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix}$$

In order to track the ZMP reference trajectory a optimal preview servo controller is designed. First, the system in Eq. 5.3 is discretized with sampling time T wich gives

$$\begin{aligned} x(k+1) &= Ax(x) + Bu(k) \\ p(k) &= Cx(k) \end{aligned} \quad (5.4)$$

with

$$\begin{aligned} x(k) &\triangleq \begin{bmatrix} x(kT) & \dot{x}(kT) & \ddot{x}(kT) \end{bmatrix}^T, \\ u(k) &\triangleq u_x(kT), \\ p(x) &\triangleq p_x(kT), \end{aligned}$$

$$\begin{aligned}
A &\triangleq \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \\
B &\triangleq \begin{bmatrix} T^3/6 \\ T^2/s \\ T \end{bmatrix}, \\
C &\triangleq \begin{bmatrix} 1 & 0 & -z_c/g \end{bmatrix}.
\end{aligned}$$

The performance index to track the reference $ZMP(k)$ is specified as

$$J = \sum_{i=k}^{\inf} Q_e e(i)^2 + \delta x^T(i) Q_x \delta x(i) + R \delta u(i) \quad (5.5)$$

where $e(i) \triangleq p(i) - p^{ref}(i)$, $Q_e, R > 0$ and Q_3 a 3×3 symmetric non-negative definite matrix. $\delta x(k) \triangleq x(k) - x(k-1)$ and $\delta u(k) \triangleq u(k) - u(k-1)$.

The optimal preview controller which minimizes the performance index defined by the Equation 5.5 is

$$u(k) = -G_i \sum_{i=0}^k e(k) - G_x x(k) - \sum_{j=1}^{N_L} G_p(j) p^{ref}(k+j) \quad (5.6)$$

where G_i, G_x and $G_p(j)$ are the gains obtained from the solution of the inverse optimal control problem. As we can observe, the controller requires to preview N_L future samples of ZMP reference trajectory to compute the current CoM sample.

5.1.3.2 Planning Center of Mass Trajectory

The motion for the center of mass is planned using the preview controller. Hence, the reference ZMP trajectory is to be provided. This reference ZMP is computed based on the footprints positions. For each basic motion segment, the human is either in double support stance or in single support stance. Particularly, for the basic motion segments LFSS_TO_DSS and LFSS_TO_DSS the robot is in single support stance, in this case the ZMP position is given the value of the footprint position of the supporting foot. Moreover, for segments DSS_TO_LFSS and DSS_TO_LFSS the robot is in double support stance. In such a case the ZMP position is given by the final footprint position of the flying foot in the previous segment.

The ZMP is to shift towards middle position of the footprints locations if the robot remains for while on this stance. Finally, using the preview controller we plan the CoM trajectory. Taking in mind that future information is required to compute the sample at each instant of

time.

5.1.4 Transfer Feet Motion to Humanoid Robot

In order to generate the joint angles trajectories of the robot HRP2, we predefined a stack of tasks and solved it by using the prioritized inverse kinematic framework. The stack of tasks consists in four tasks:

- Homogeneous transformation task for each left foot, position and orientation, or homogeneous transformation task for each right foot, position and orientation,
- Center of Mass task, only projection to the ground, X and Y positions,
- Orientation vector task for the chest.

These tasks are defined because first it is required to ensure feet contact with the ground, then second task is to ensure the CoM projection trajectory and the third task is to kept the body upright.

The first tasks are used to move the feet of the robot, the second task is used to manage balance and the four task is used the keep the chest of the robot vertical. The target inputs for tasks 1) and 2) are computed as stated in subsection 5.1.2. Then the joint angles trajectories for the robot is computed for the human motion sequence, and finally played on the robot. In practice, the number of samples to stabilize the preview controller corresponds to 1.6 seconds of motion. This means that it is required to know in advance some quantity of motion. In fact, we have increased it to 2.0 seconds of motion to ensure the stability of the preview controller.

5.2 Online Human Feet Motion Imitation

The online human motion imitation by humanoid robot problem consists fo capturing human motion online, generate robot motion online, and sent it to the robot. In Chapter 4 we detailed a framework to imitate human motion by a humanoid under the hypothesis that the feet of the robot were fixed to the ground. The flowchart of data start with the capturing human motion, next generating robot motion, and finally the performance of the robot. At each step only the current and previous samples of motion were used to generating robot motion.

In the case of including feet motion this approach of using only the current and previous samples cannot be used, because when a robot moves its feet the balance cannot be guaranteed. Current methods to balance robot not allow to treat this problem online, it is required to observe part of the motion first, next analyzed it, and finally plan feet and CoM motion that guarantees the balance of the robot to avoid a fall.

We use this strategy under the hypothesis that the robot should observe first part of the motion and then plan its own motion to imitate it. The general idea is the following, at the

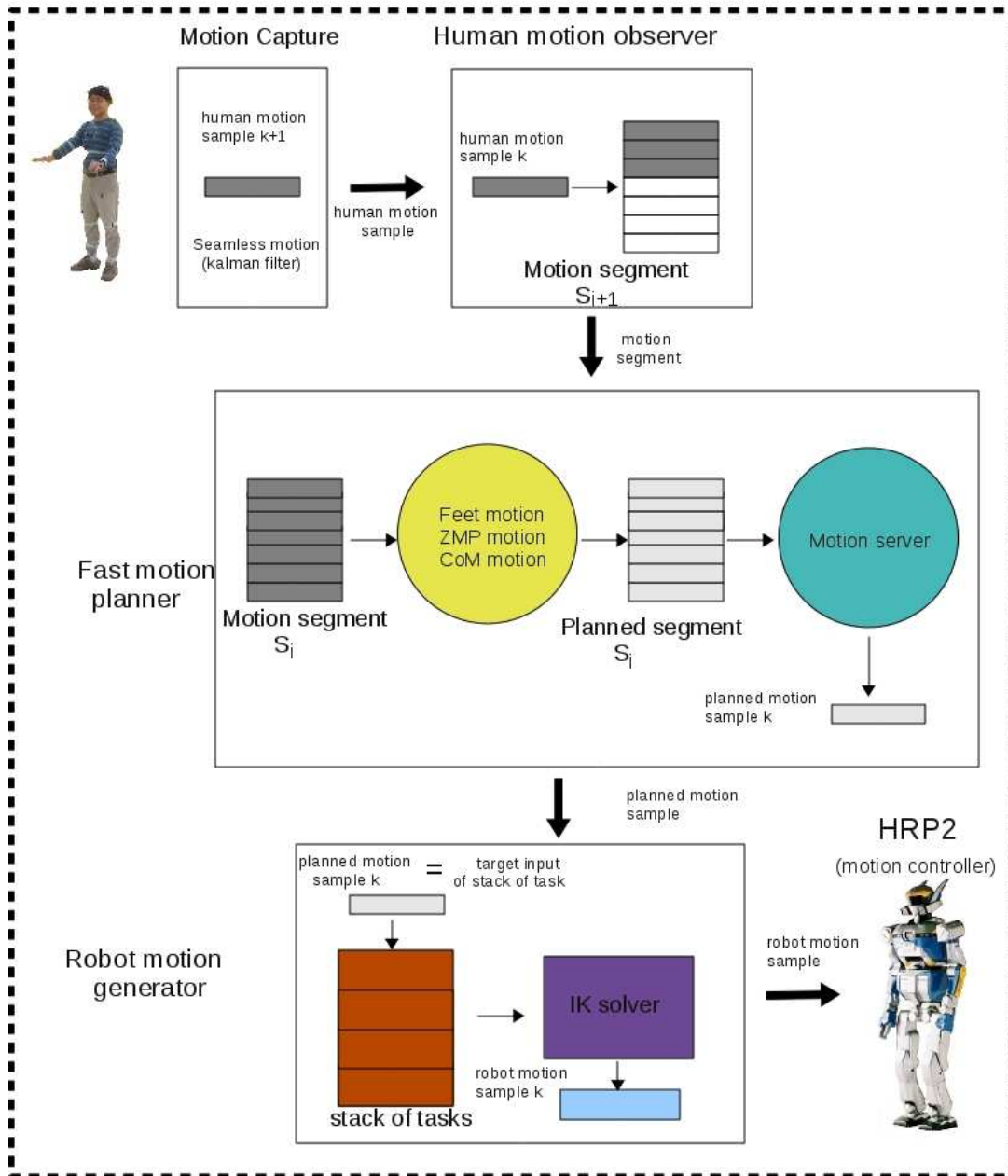


Figure 5.6: Framework used for online motion transfer. From human motion capture to HRP2.

beginning the robot observes a segment of N samples of human motion (S_0), at the instant the N sample arrives the robot plans its own motion and start executes it. The planned robot motion has the same number of samples as the observed one. At the same time the robot is executing its planned motion, it continues observing the human. At the moment the robot has played the N sample of human motion, the robot has observed another segment with N samples (S_1). Again, the robot plans its motion to imitate this new segment and executes it. This loop of observing, planning, executing motion provides the framework of online imitation.

The fact that both the planned and the observed motion has the same number of samples, provides synchronization between observing and executing motion. In more detail, when the robot is observing the k -th sample of segment S_i , the robot is executing its k -th planned sample of segment S_{i-1} .

Figure 5.6 show the framework for online transferring human motion to robot humanoid including feet. There are five stages:

1. Human motion capture: the human motion is captured online, the markers are cleaned by the filter Kalman as explained in Section 4.2,
2. Human motion observer: N samples of motion are recorder to form a segment of motion,
3. Fast motion planner: the feet, ZMP and CoM trajectories are planned, additionally a server is used to send planned samples at constant rate,
4. Robot motion generator: uses the planned samples as target inputs to the stack of tasks. The robot motion is computed by solving this stack of task,
5. Robot execution: the robot executes the imitation,

5.2.1 Human Motion Segments

The human motion is analyzed by segments of a constant number samples. The analysis consists to recover the minimal information to start planning motion. For each segment height feet are analyzed to plan CoM and feet trajectories. The difficulties in this segmentation is that for each segment the information of the trajectory is incomplete. For example, assume we observe N samples every 2 seconds and are analyzing the segment from 2.0 s to 4.0 s. We plan when the N sample arrives. This case is illustrated in Figure 5.7, we observe that this segment comprises the basic segments S_1 , S_2 and S_3 . The plan for S_1 , S_2 and S_3 is to be achieved. However, S_3 is a basic segment broken, so that in a real situation we don't know what motion follows. Furthermore, we have to plan motion even for this incomplete segment in order to maintain the synchronization between observing and executing motion. The solution to this problem is developed in the rest of this section.

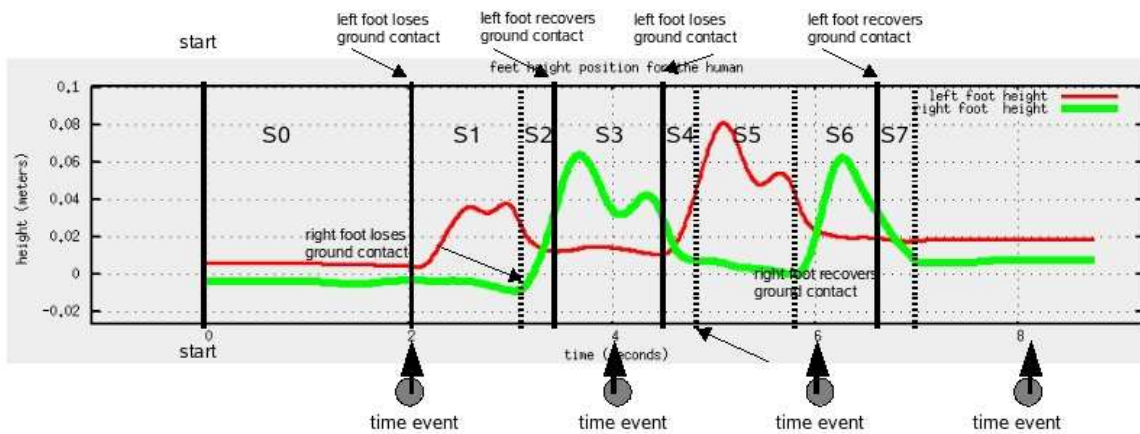


Figure 5.7: Plot of feet height indicating the instant occurs stance change, and marks basis segments. Additionally, it is marked where the motion should be computed, in almost all cases there are incomplete segments when motion should be computed.

5.2.2 Planning Feet Motion

Every N observed samples of motion we have to plan. The segment can have several basic segments, although either one or two of them can be broken at the beginning or at the end of the segment. Consequently, next segment starts with the next part of this broken basic segment. To plan the feet trajectory we split it into position and orientation. The position is analyzed in two parts, first the horizontal trajectory (x, y) and next the height z . We observe from Figure 5.7 that only the trajectory for one foot is broken, the foot flying, while the other remains fixed. For this reason, we only analyze the flying foot.

The horizontal trajectory of the broken segment is planned by using via points of the observed motion, see Figure 5.8. These points are then interpolated using a third order polynomial to generate a smooth curve, with this polynomial we can set zero final speed to reduce feet impact. The height is planned in a similar way, but the via points are tested to not exceed the maximum height allowed for the robot. Thus, the via points are limited to this maximum, see Figure 5.9. As we mentioned previously, the yaw rotation is considered for the foot orientation. The plan of this orientation is achieved in the same way as for the horizontal trajectory.

Once the next N samples has been observed, we plan for the next part of the broken segment and other basic segments detected. To plan for this new segment, again we interpolate between via points of position and orientation. But, in this case the complete basic segment is considered. In fact, we interpolate using not only points of the current observed motion, but we include some of the previous one, see Figures 5.10 and 5.11. The height trajectory is critical because it determines the speed of the foot at landing instant. We modify the trajectory if it does not have zero or close to zero velocity, particularly we compute a mini mun jerk trajectory from the last sample in the previous planned segment to zero. The yaw trajectory is planned in the same way

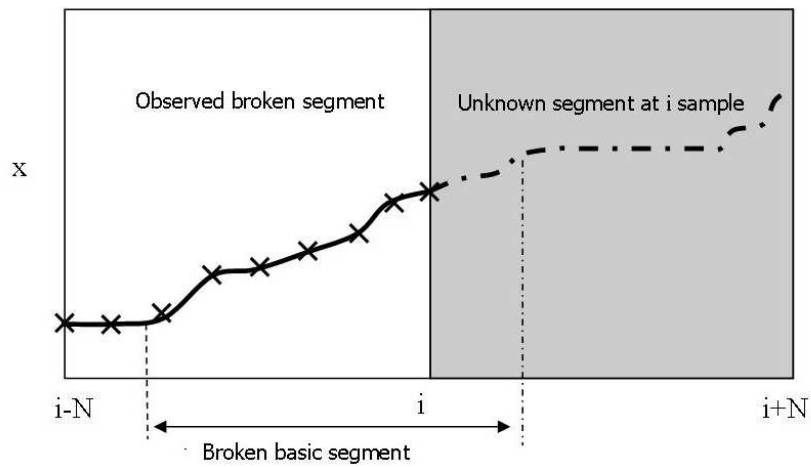


Figure 5.8: Planning the x component of the horizontal trajectory. The via points are represented by cross marks, the observed segment (solid line) is in the white region, while the next segment is in the gray region (dashed line).

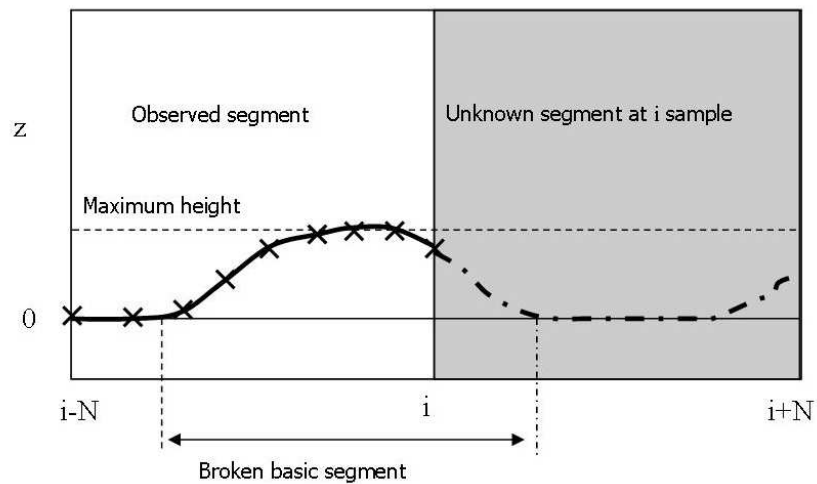


Figure 5.9: Planning the height Z trajectory. The via points are represented by cross marks, the observed segment (solid line) is in the white region, while the next segment is in the gray region (dashed line). The thin-horizontal-dashed line indicates the maximum allowed height.

as the horizontal trajectory.

The other basic segments are planned as explained in subsection 5.1.2. The planning of the trajectories for the other foot follows the same procedure explained above. But, as mentioned earlier when one foot is flying the other is fix to the ground. So the plan for the supporting foot is very easy it is constant.

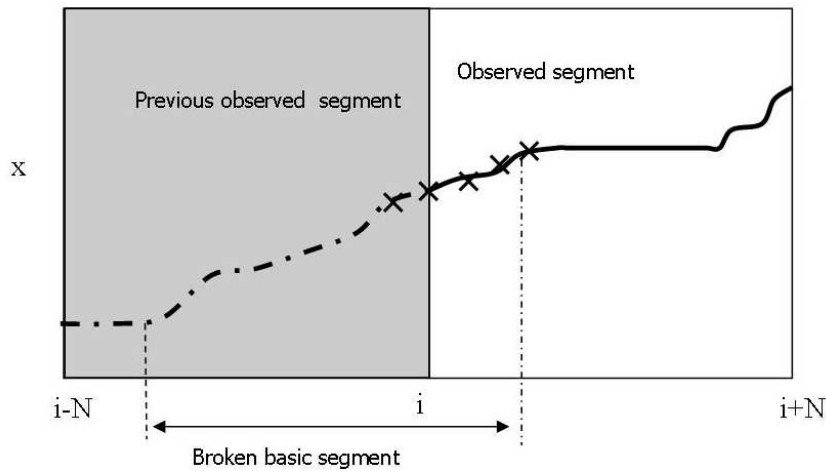


Figure 5.10: Planning the x component of the horizontal trajectory. To plan this trajectory we use the information of the basic segment. The via points are represented by cross marks, the observed segment (solid line) is in the white region, while the next segment is in the gray region(dashed line).

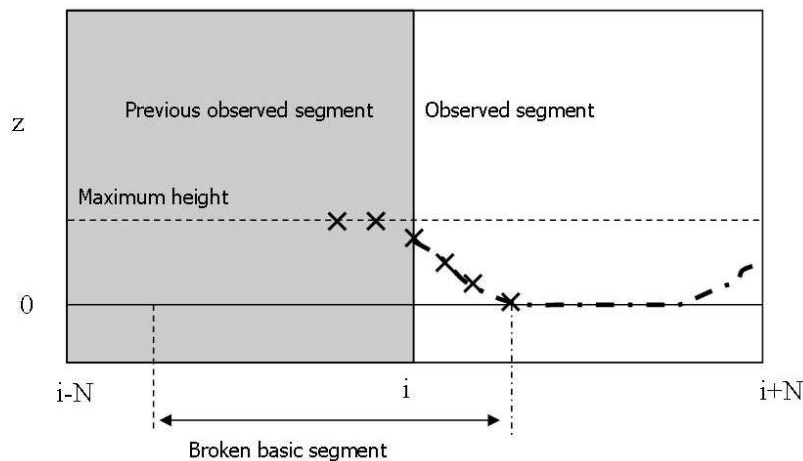


Figure 5.11: Planning the height Z trajectory. To plan this trajectory we use the information of the basic segment. The via points are represented by cross marks, the observed segment (solid line) is in the white region, while the next segment is in the gray region(dashed line). The thin-horizontal-dashed line indicates the maximum allowed height.

5.2.3 Planning Center of Mass Motion

The preview controller allows to determine the CoM projected trajectory from a given ZMP reference trajectory, but to compute the current position of the CoM it is required to observe in advance a number of samples of ZMP. So our problem now is to compute the reference ZMP trajectory.

In order to define a reference ZMP trajectory, the main idea is to shift the ZMP position on current support foot. In fact, the ZMP is only modified at the instant a basic segment ends. That is, the ZMP remains constant until the instant the flying foot lands, which cannot be known at in advance. From this planned ZMP, the CoM trajectory can be computed using the preview controller. Figure 5.12 shows two plots. The upper plot shows the height trajectories of feet and the instants at which ZMP changes, denoted by the vertical lines between plots. The gray region indicates the unknown segment. The lower plot shows the ZMP (y component) trajectory at the instant it changes. It can be seen that the ZMP remains constant in the gray region. That is to plan, we need the final value of ZMP in the previous segment, so no more information is required to plan. It is possible that the instant at which N samples has been observed is in the double support. If such a situation occurs we keep the ZMP constant, as can be seen from the picture. In the case the double support has a large number of samples, the ZMP is shift to the center point of the feet positions.

5.2.4 Transfer Feet Motion to Humanoid

To transfer the human motion to the humanoid robot we use our inverse kinematic solver and a predefined stack of tasks; including feet transformation tasks, CoM task and vector orientation task for the chest. The target for each task is provided by a fast planner device, see Figure 5.6. Human motion segment is the input for this planner device. This device implements the planning for feet trajectories, ZMP trajectory and CoM trajectory. Moreover, this devices implements a motion server which is in charge of sending planned samples (CoM and feet trajectories) at constant rate to the robot motion generator. The robot motion generator uses the predefined stack of tasks, the planned sample as target for the stack, and inverse kinematics to generate the robot configurations that imitate the human feet. Finally, these configurations are sent to the robot controller interface via a Genom plugging, refer to subsection 4.5.1.

5.3 Imitation Including Stepping

In order to imitate the whole body, we use the same framework as imitating feet motion, but in the fast planner we plan also the upper body. In this case a method is required to make compatible the footprints position and orientation with the upper body.

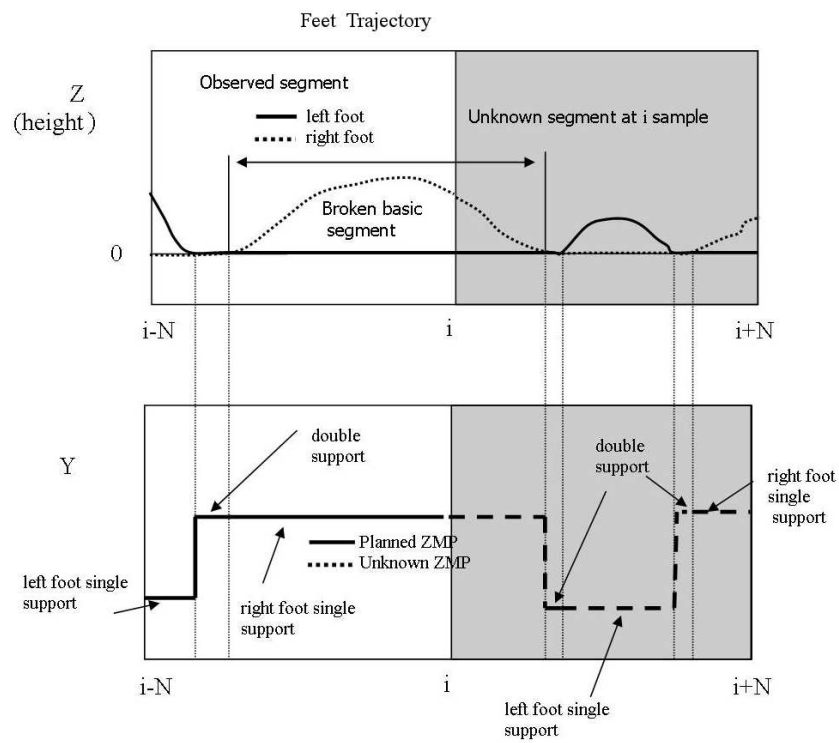


Figure 5.12: Planning the ZMP trajectory from the feet positions. To plan the ZMP it is only required the the position of the last landing foot. Attention must be made on the feet height, they determines when the ZMP must be changed.

5.3.1 A Complete Humanoid Normalized Model

A complete Humanoid Normalized model to transfer human motion to the robot includes,

$$[P_h, V_h, N_c, V_w, P_{lh}, N_{la}, P_{rh}, N_{ra}, P_{lf}, V_{lf}, P_{rf}, V_{rf}, CoM] \quad (5.7)$$

where P_h is a point representing the position of the head, V_h is a vector representing the orientation of the head, N_c is a vector representing the normal of the chest plane, V_w is a vector representing the orientation of the waist, P_{lh} is a point representing the position of the left hand, N_{la} is a vector representing the normal to the left arm plane, P_{rh} is a point representing the position of the right hand, N_{ra} is a vector representing the normal to the right arm plane, P_{lf} is a point representing the position of the left foot, V_{lf} is a vector representing the orientation of the left foot, P_{rf} is a point representing the position of the right foot, V_{rf} is a vector representing the orientation of the right foot, and CoM is a point representing the center of mass projection to the ground. This representation is generic and free of kinematic model. That is, in order to transfer this motion representation it is not required to save any kinematic hierarchy or body segment lengths. The only parameter to save is the height of the subject who performed the motion capture session.

5.3.2 Human Motion Transfer Including Stepping

To transfer the human motion we use the complete normalized model, we compute elements associated to the head, chest, waist, arms as developed in Chapter 3. The CoM and feet motion of the HN-model is computed as stated in the previous sections.

At this point the upper body and feet are not compatible. That is the orientation of the upper body and the lower body do not fit because of the limitation of the robot to realize all the human step configurations. In this case we could use the following strategy: keep the foot prints and modify the upper body properties by translating and rotating them. This is computed at the planning stage. The framework to transfer human motion including stepping is the same as the framework of imitating stepping except for the HN-model representation. The general idea remains the same.

5.4 Conclusion

In this Chapter we have developed and discussed the strategies to implement online feet motion imitation. The key point is the strategies used to cope with incomplete information. The final human motion representation including stepping is generic that it allows to transfer motion to characters or humanoid robot. The main advantage being that is free of kinematic model. The

proposed framework requires to observe in advance a predefined number of samples in order to start imitating human motion. Mainly, because of the balance problem of the robot.

6

Conclusion and Perspectives

6.1 Conclusion

In this work we have studied the problem of transferring human motion to a humanoid robot. We called this process imitation, because the robot observes human motion, represents it, plans its own motion to act accordingly, and realises a performance. In the context of imitation our interest is limited to mimic human motion to some extent. We are not interested in learning or understanding the motion. However, the choice of representation of motion is very important in those kind of studies, as reported by Kulic et al. [Kulic and Nakamura 2009]. In this way, we adopted a representation of motion that is inspired from studies in Neuroscience Research and Computer Animation area.

The Humanoid-Normalized model was developed with the objective of being generic and serve as vehicle to transfer motion to a humanoid. This representation involves meaningful information about the motion properties, a difference with representations based on joint trajectories . We use mainly two kinds of information: positions and orientation vectors. With this representation we obtain directly the information of the hands, the feet and the head trajectories. Information of the orientation of some body parts are also directly available. In particular information of the chest, the head and waist. That is a big difference respect to representations baaes on joint trajectories, where is required to know a proper kinematic model to give sense of motion to joint trajectories. What is more, in motion capture systems same marker set but using different kinematic models produces significantly differences in the trajectories for the common

degrees of freedom. Thus, resulting in a particular representations of motion. That is true, that retargeting methods are available to solve this kind of problem but results again in specific joint trajectories to a kinematic model.

However, in our representation a inverse kinematic solver is required in order to reproduce the motion on a digital character or a robot. This is not a big problem, because generic prioritized inverse kinematics solvers has been well studied. At this point, the redundancy of each particular kinematic model is exploited to reproduce the motion more accurately.

The human motion representation we adopted is free of kinematic model. The kinematic model and an inverse kinematics solver are used in the case we require to produce motion on a digital character or humanoid. Although this representation is generic, there are some issues that have to be studied like the scaling of marker positions and the feet motion. This representation can be improved to produce more accurately human-like motion by incorporating more information like the anticipation nature of the head and the coordination between head, chest and waist, or local behaviours. To say a local behavior, when humans are stand-up the head follows circular trajectories, which differs significantly of the usual static stand-up postures in humanoids. This model inspired from neuroscience research opens up new windows towards incorporating biological principles in humanoid motion control [Berthoz 2000], [Sreenivasa et al. 2009].

Another point of interest it to evaluate the similarity between motion. This is issue is crucial for motion learning because such metrics are to be evaluated. In motion imitation two criteria are used a qualitative and a quantitative. The qualitative measure use people point of view to evaluate it. This produces a dependency on the criteria and observation of people. To recall the example in the introduction where the child is imitating the robot, I personally qualify the movement of the child in respect of the context as imitation. However, the posture of the leg is quite different to the posture of the robot at that moment, what is more the robot was lifting-up its left leg while the child lifted-up his right leg. The quantitative evaluation, however, requires to assign a value, the usual criteria is based on the root mean square (RMS) on the joint trajectories. We adopt the same index but to evaluate each property in our representation. We have found this index based on motio properties more adequate to evaluate motion properties than the one based joint trajectories.

The human motion transfer framework we developed is generic in respect of the target kinematic model, either a character or humanoid. This is a direct consequence of the employed motion representation. The kinematic model is required at the moment of generating robot motion. In the case of transferring motion to humanoid robot, the main advantage comes from the inverse kinematic approach to generate robot motion which provides solution at 20 ms, fast enough to imitate a large range of human motions. On the other hand, the main drawback comes from the inverse kinematic approach too. Because the inertial forces and the gravity forces are not considered, the stability of the robot is hard to achieve for fast movements even

if the stabilizer of the robot is turned on.

The limitation to transfer fast motions is the main disadvantage of using a kinematic based approach, however the generation of motion is quite fast for real-time implementations. Even if we use a center of mass trajectory to balance, it does not consider the inertial forces that appears at high speed motions. The robot can fall if the inertial forces are important.

On the other hand, We can see the human motion imitation in other sense, that is, the human is controlling the robot with his body, he is teleoperating the robot. However, we do not consider this is our case, because teleoperation requires high precision in the task. This is not our main interest, even if we are able to control the robot to reach some objects or to control it from a small mannequin. However, this is an interesting point to investigate. A biped robot teleoperated by the human body.

The inclusion of stepping in the online transfer framework arises the stability problem. The dynamic nature of the stepping claims to include a dynamic model to cope with this issue. In our case we use the table-car model of the robot and the preview controller proposed by Kajita et al. [Kajita et al. 2003], however to stabilize the robot when stepping it is required to know future information. This is overcome if a delay is intentionally allowed. Furthermore, feet trajectories are unknown so that frequently it is required to plan. In this situation, the priority between motion and stability is obvious. The robot must maintain its stability first to be able to perform other kind of movements. *The final conclusion on this issue are to be added.*

6.2 Perspectives

Markerless human motion transfer was one of the last attempts during this work. The main idea is to replace the motion capture system by a specialized vision system capable of estimate the motion properties included in the Humanoid Normalized model. In the RAP research group, work lead by Frédéric Lerasle, a first version of a system capable of determine the motion properties related to the arms is available, Figure 6.1. This system could evolve to include all the other properties even the feet motion.

Humanoid tele-operated by human body is one of the ultimate goals in robotic research. However, it requires high precision in the execution of a goal task and also requires interaction with the environment in real-time. Our work can be the base to start working on these issues.

Introducing contact motion analysis and dynamics can allow us to interact with the environment and to produce faster movements. The challenge in this case is to cover these issues in real-time.



Figure 6.1: A snapshot of the motion capture system based on stereo vision and particle filtering.



Motion Capture and Marker Set for our Experiments

The motion capture system we use to capture human motion is equipped with 10 infra-red tracking cameras (MotionAnalysis, CA, USA). The system is capable of tracking the position of markers within a 5x5 m space within an accuracy of 1mm, at a data rate of 100 Hz. A picture of the capture room, Grand Salle, in the Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS, is illustrated in Figure A.1.



Figure A.1: The Grand Salle is housing for the robots available at LAAS-CNRS where also is installed the motion capture system.

The marker set we used in our experiments is composed of 41 markers and 60 links between them. An image of a frame showing the markers and links, taken from the panel-view of the software accompanying the motion capture system, is shown in Figure A.2

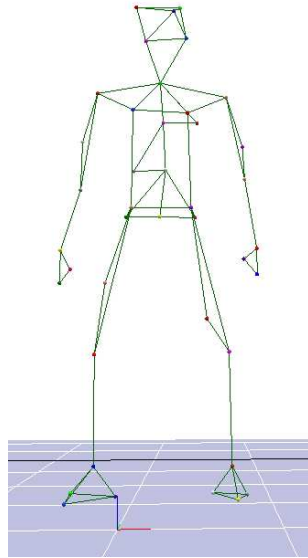


Figure A.2: Frame for the marker set used showing the 41 markers and 60 links.

A more detailed description of the placement of marker model is depicted in Figures A.3, A.4, A.5, A.6 . Among all these markers, only 21 are used in our experiments. The rest of the marker are used to have a more robust marker set when transferring motion online. Furthermore, we use a Linear Kalman Filtering to predict the position of the markers in such a case it was lost.

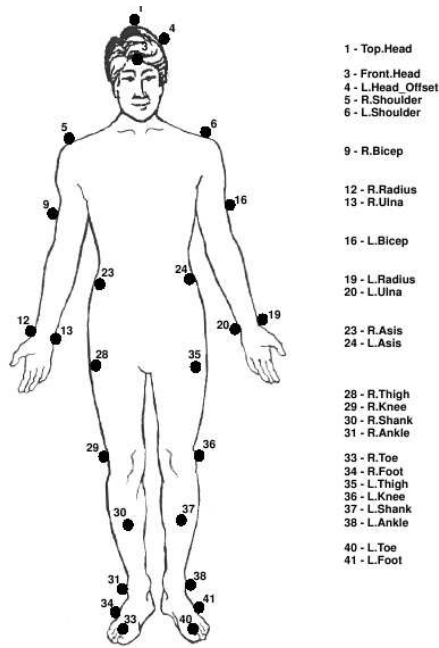


Figure A.3: Marker set front view.

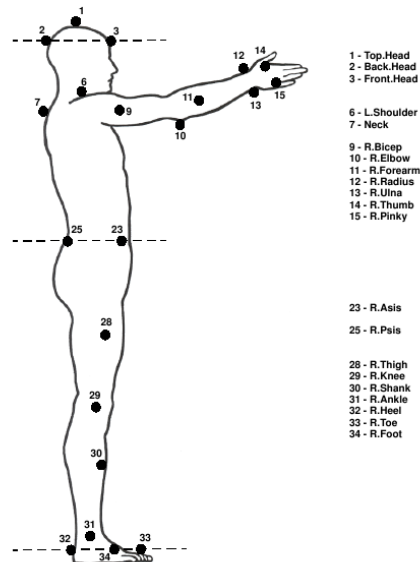


Figure A.4: Marker set right side view.

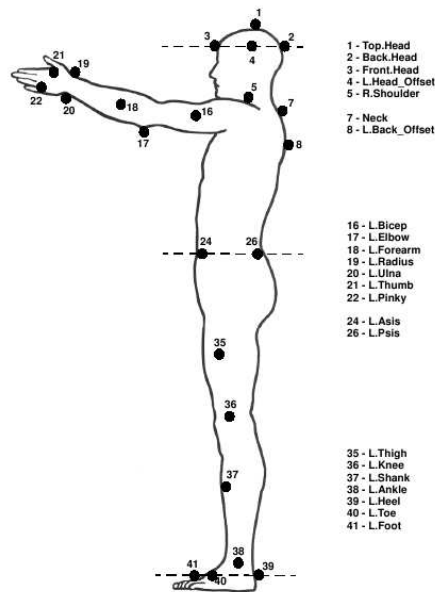


Figure A.5: Marker set left side view.

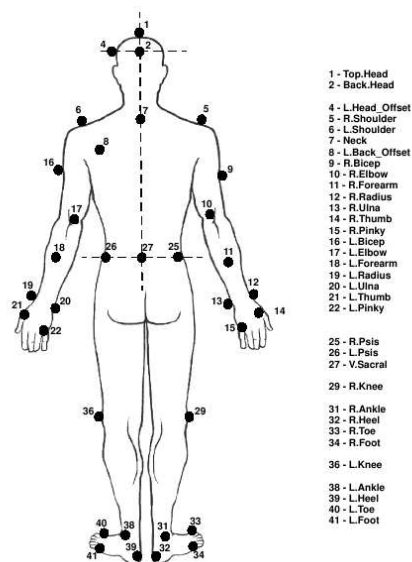


Figure A.6: Marker set rear view.

References

- BAERLOCHER, P. AND BOULIC, R. 1998. Task-priority formulations for the kinematic control of highly redundant articulated structures. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vol. 1. 323–329 vol.1.
- BAKER, R. 2007. The history of gait analysis before the advent of modern computers. *Gait & Posture* 26, 3, 331–342.
- BERTHOZ, A. 2000. *The brain's sense of movement*. Harvard University Press, Cambridge, MA.
- BILLARD, A., CALINON, S., AND GUENTER, F. 2006. Discriminative and Adaptive Imitation in Uni-Manual and Bi-Manual Tasks. *Robotics and Autonomous Systems* 54, 5, 370–384.
- CHOI, K. J. AND KO, H. S. December 2000. Online motion retargeting. *The Journal of Visualization and Computer Animation* 11, 5, 223–235.
- DARIUSH, B., GIENGER, M., ARUMBAKKAM, A., GOERICK, C., ZHU, Y., AND FUJIMURA, K. 2008. Online and markerless motion retargeting with kinematic constraints. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 191–198.
- DARIUSH, B., GIENGER, M., JIAN, B., GOERICK, C., AND FUJIMURA, K. 2008. Whole body humanoid control from human motion descriptors. In *Proceedings of the IEEE International Conference on Robotics and Automation*. 2677–2684.
- DAVID, A., BRUNEAU, O., AND J.G., F. 2005. *Climbing and Walking Robots*. Springer, Chicago, Chapter Dynamic Stabilization of an Under-Actuated Robot Using Inertia of the Transfer Leg, 551 – 559.
- ESTEVEZ JARAMILLO, C. E. 2007. Motion planning: from digital actors to humanoid robots. Ph.D. thesis, Institut National Polytechnique de Toulouse, LAAS-CNRS.
- FLEURY, S., HERRB, M., AND CHATILA, R. 1997. Genom: A tool for the specification and the implementation of operating modules in a distributed robot architecture. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 842–848.
- FORET, J., BRUNEAU, O., AND FONTAINE, J. 2003. Unified approach for m-stability analysis and control of legged robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vol. 1. 106 – 111 vol.1.
- GARCIA, M., CHATTERJEE, A., RUINA, A., AND COLEMAN, M. 1998. The simplest walking model: Stability, complexity, and scaling. *Journal of Biomechanical Engineering* 120, 2, 281–288.

- GLEICHER, M. 1998. Retargetting motion to new characters. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM, New York, NY, USA, 33–42.
- GOSWAMI, A. 1999. Postural Stability of Biped Robots and the Foot-Rotation Indicator (FRI) Point. *The International Journal of Robotics Research* 18, 6, 523–533.
- GOSWAMI, A. AND KALLEM, V. 2004. Rate of change of angular momentum and balance maintenance of biped robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*. Vol. 4. 3785 – 3790.
- GUENTER, F., HERSCH, M., CALINON, S., AND BILLARD, A. 2007. Reinforcement Learning for Imitating Constrained Reaching Movements. *RSJ Advanced Robotics, Special Issue on Imitative Robots* 21, 13, 1521–1544.
- HERSCH, M., GUENTER, F., CALINON, S., AND BILLARD, A. 2008. Dynamical system modulation for robot learning via kinesthetic demonstrations. *IEEE Transactions on Robotics* 24, 6 (dec.), 1463 –1467.
- HIROSE, M. AND OGAWA, K. 2007. Honda humanoid robots development. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 365, 1850, 11–19.
- ILL-WOO, P., JUNG-YUP, K., JUNGHO, L., AND OH, J.-H. 2005. Mechanical design of humanoid robot platform khr-3 (kaist humanoid robot 3: Hubo). In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*. 321 –326.
- INAMURA, T., NAKAMURA, Y., EZAKI, H., AND TOSHIMA, I. 2001. Imitation and primitive symbol acquisition of humanoids by the integrated mimesis loop. In *Proceedings of the IEEE International Conference on Robotics and Automation*. Vol. 4. 4208–4213 vol.4.
- ISHIDA, T., KUROKI, Y., AND YAMAGUCHI, J. 2003. Mechanical system of a small biped entertainment robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vol. 2. 1129 – 1134 vol.2.
- JONES, S. 2006. Infants learn to imitate by being imitated. In *International Conference on Development and Learning*.
- KAGAMI, S., NISHIWAKI, K., SUGIHARA, T., KUFFNER, J., INABA, M., AND INOUE, H. 2001. Design and implementation of software research platform for humanoid robotics: H6. In *Proceedings of the IEEE International Conference on Robotics and Automation*. Vol. 3. 2431 – 2436 vol.3.
- KAJITA, S., HIRUKAWA, H., HARADA, K., AND YOKOI, K. 2009. *Introduction à la commande des robots humanoïdes*. Springer.
- KAJITA, S., KANEHIRO, F., KANEKO, K., FUJIWARA, K., HARADA, K., YOKOI, K., AND HIRUKAWA, H. 2003. Biped walking pattern generation by using preview control of zero-moment point. In *Proceedings of the IEEE International Conference on Robotics and Automation, 2003*. Vol. 2. 1620 – 1626 vol.2.
- KAJITA, S., TANI, K., AND KOBAYASHI, A. 1990. Dynamic walk control of a biped robot

- along the potential energy conserving orbit. In *Proceedings of the IEEE International Workshop on Intelligent Robots and Systems*. 789–794 vol.2.
- KANEHIRO, F., SULEIMAN, W., LAMIRAUX, F., YOSHIDA, E., AND LAUMOND, J.-P. 2008. Integrating dynamics into motion planning for humanoid robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 660–667.
- KANEKO, K., KANEHIRO, F., KAJITA, S., HIRUKAWA, H., KAWASAKI, T., HIRATA, M., AKACHI, K., AND ISOZUMI, T. 2004. Humanoid robot hrp-2. In *Proceedings of the IEEE International Conference on Robotics & Automation, 2004*.
- KAYE, K. 1982. *The mental and social life of babies: how parents create persons*. University of Chicago Press, Chicago.
- KIM, S., KIM, C., YOU, B., AND OH, S. 2009. Stable whole-body motion generation for humanoid robots to imitate human motion. In *Proceeding of IEEE-RAS International Conference on Intelligent Robots and Systems*,. 2518–2524.
- KULIC, D. AND NAKAMURA, Y. 2009. Comparative study of representations for segmentation of whole body human motion data. In *Proceedings of the IEEE/RSJ International conference on Intelligent Robots and Systems*. IEEE Press, Piscataway, NJ, USA, 4300–4305.
- LIÉGEOIS, A. 1977. Automatic supervisory control of the configuration and behavior of multi-body mechanisms. *IEEE Transactions on Systems, Man and Cybernetics* 7, 12 (Dec.), 868–871.
- LIM, H.-O. AND TAKANISHI, A. 2007. Biped walking robots created at Waseda University: WL and WABIAN family. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 365, 1850, 49–64.
- MACIEJEWSKI, A. A. AND KLEIN, C. A. 1985. Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments. *The International Journal of Robotics Research* 4, 3, 109–117.
- MARIN-URIAS, L., SISBOT, E., PANDEY, A., TADAKUMA, R., AND ALAMI, R. 2009. Towards shared attention through geometric reasoning for human robot interaction. In *Proceeding of 9th IEEE-RAS International Conference on Humanoid Robots*,. 331–336.
- MCGEER, T. 1990. Passive dynamic walking. *International Journal of Robotics Research* 9, 2, 62–82.
- MEDVED, V. 2001. *Measurement of Human Locomotion*. CRC Press LLC.
- MULTON, F., KULPA, R., AND BIDEAU, B. 2008. Mkm: A global framework for animating humans in virtual reality applications. *Presence: Teleoper. Virtual Environ.* 17, 1, 17–28.
- MUYBRIDGE, E. 1979. *Muybridge's Complete Human and Animal Locomotion*. Dover Publications.
- NAKAMURA, Y. 1991. *Advanced Robotics: Redundancy and Optimization*. Addison-Wesley Longman Publishing, Boston.
- NAKAMURA, Y. AND HANAFUSA, H. 1986. Inverse kinematics solutions with singularity robustness for robot manipulator control. *Journal of Dynamic Systems, Measurement, and Control*. 108, 3, 163–171.

- NAKAOKA, S., NAKAZAWA, A., YOKOI, K., HIRUKAWA, H., AND K., I. 2003. Generating whole body motions for a biped humanoid robot from captured human dances. In *Proceedings of the IEEE International Conference on Robotics and Automation*.
- O'BRIEN, J. F., BODENHEIMER, R. E., BROSTOW, G. J., AND HODGINS, J. K. 2000. Automatic joint parameter estimation from magnetic motion capture data. In *Proceedings of Graphics Interface 2000*. 53–60.
- OGURA, Y., AIKAWA, H., SHIMOMURA, K., MORISHIMA, A., HUN-OK, L., AND TAKANISHI, A. 2006. Development of a new humanoid robot wabian-2. In *Proceedings of the IEEE International Conference on Robotics and Automation*. 76–81.
- PATLA, A., ADKIN, A., AND BALLARD, T. 1999. Online steering: coordination and control of body center of mass, head and body reorientation. *Experimental Brain Research* 129, 4, 629–634.
- PEER, A., HIRCHE, S., WEBER, C., KRAUSE, I., AND BUSS, M. 2008. Intercontinental cooperative telemanipulation between germany and japan. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2715–2716.
- RICH, C., PONSLEUR, B., HOLROYD, A., AND SIDNER, C. L. 2010. Recognizing engagement in human-robot interaction. In *Proceeding of the 5th ACM/IEEE international conference on Human-robot interaction*. ACM, New York, NY, USA, 375–382.
- RIZZOLATTI, G. AND CRAIGHERO, L. 2004. The mirror-neuron system. *Annual Review of Neuroscience* 27, 169192.
- ROSENHAHN, B., KLETTE, R., AND METAXAS, D. 2008. *Human Motion: Understanding, Modelling, Capture, and Animation (Computational Imaging and Vision)*. Springer.
- RUCHANURUCKS, M., NAKAOKA, S., KUDOH, S., AND IKEUCHI, K. 2006. Humanoid robot motion generation with sequential physical constraints. In *Robotics and Automation, Proceedings IEEE International Conference on*. 2649–2654.
- SAINT-AIME, S., LE PEVEDIC, B., LETELLIER-ZARSHENAS, S., AND DUHAUT, D. 2009. Emi - my emotional cuddly companion. In *Proceeding of IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN*. 705–710.
- SAKAGAMI, Y., WATANABE, R., AOYAMA, C., MATSUNAGA, S., HIGAKI, N., AND FUJIMURA, K. 2002. The intelligent asimo: system overview and integration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vol. 3. 2478 – 2483 vol.3.
- SCHAAL, S. 1999. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences* 6, 233–242.
- SICILIANO, B. AND SLOTINE, J. 1991. A general framework for managing multiple tasks in highly redundant robotic systems. In *Proceedings of the IEEE International Conference on Advanced Robotics*. 1211–1216.
- SREENIVASA, M.-N., SOUERES, P., LAUMOND, J.-P., AND BERTHOZ, A. 2009. Steering a humanoid robot by its head. In *Intelligent Robots and Systems, IROS 2009. IEEE/RSJ International Conference on*.

- SUGIHARA, T., NAKAMURA, Y., AND INOUE, H. 2002. Real-time humanoid motion generation through zmp manipulation based on inverted pendulum control. In *Proceedings of the IEEE International Conference on Robotics and Automatio*. Vol. 2. 1404 – 1409 vol.2.
- SULEIMAN, W., YOSHIDA, E., KANEHIRO, E., LAUMOND, J.-P., AND MONIN, A. 2008. On human motion imitation by humanoid robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*.
- SVEISTRUP, H., SCHNEIBERG, S., MCKINLEY, P., MCFADYEN, B., AND LEVIN, M. 2008. Head, arm and trunk coordination during reaching in children. *Experimental Brain Research* 188, 2, 237–247.
- TAKANO, W., YAMANE, K., SUGIHARA, T., YAMAMOTO, K., AND NAKAMURA, Y. 2006. Primitive communication based on motion recognition and generation with hierarchical mimesis model. In *Proceedings of the IEEE International Conference on Robotics and Automation*. 3602–3609.
- TANIE, K. 2003. Humanoid robot and its application possibility. In *Proceedings of the IEEE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems, MFI*. 213–214.
- UDE, A., ATKESON, C., AND M., R. 2004. Programming full-body movements for humanoid robots by observation. In *Robotics and Autonomous Systems*. Vol. 47. 93–108.
- VUKOBRATOVIC, M. AND STEPANENKO, J. 1972. On the stability of anthropomorphic systems. *Mathematical Biosciences* 15, 1–37.
- WAMPLER, II, C. W. 1986. Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Trans. Syst. Man Cybern.* 16, 1, 93–101.
- WEBER, W. E. AND WEBER, E. 1991. *Mechanics of the human walking apparatus*. Springer-Verlag.
- WHITNEY, D. 1969. Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on Man Machine Systems* 10, 47–53.
- WILKE, L., CALVERT, T., RYMAN, R., AND FOX, I. 2005. From dance notation to human animation: The labandancer project: Motion capture and retrieval. *Comput. Animat. Virtual Worlds* 16, 3-4, 201–211.
- WOONG, K., KIM, H., JOONG KYUNG, P., CHANG HYUN, ROH ABD JAWOO, L., JAEHO, P., WON-KUK, K., AND KYUNGSHIK, R. 2007. Biped humanoid robot mahru iii. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*. 583 –588.
- YAMANE, K. AND HODGINS, J. 2009. Simultaneous tracking and balancing of humanoid robots for imitating human motion capture data. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2510–2517.
- YAMANE, K. AND NAKAMURA, Y. 2003. Dynamics filter - concept and implementation of online motion generator for human figures. *IEEE Transactions on Robotics and Automation* 19, 3 (June), 421–432.

- YOSHIDA, E., KANOUN, O., ESTEVES, C., AND LAUMOND, J.-P. 2006. Task-driven support polygon humanoids. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*.
- YOSHIDA, E., MALLET, A., LAMIRAUX, F., KANOUN, O., STASSE, O., POIRIER, M., DOMINEY, P.-F., LAUMOND, J.-P., AND YOKOI, K. 2007. give me the purple ball” - he said to hrp-2 n.14. In *Proceedings of the IEEE/RAS International Conference on Humanoid Robots*. 89–95.
- YOSHIDA, E., POIRIER, M., LAUMOND, J.-P., ALAMI, R., AND YOKOI, K. 2007. Pivoting based manipulation by humanoids: a controllability analysis. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1130–1135.
- YU, T., SHEN, X., LI, Q., AND GENG, W. 2005. Motion retrieval based on movement notation language: Motion capture and retrieval. *Comput. Animat. Virtual Worlds* 16, 3-4, 273–282.
- ZAIER, R. AND KANDA, S. 2007. Piecewise-linear pattern generator and reflex system for humanoid robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*. 2188 –2195.
- ZHAO, J. AND BADLER, N. I. 1994. Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Trans. Graph.* 13, 4, 313–336.